

fig. 1

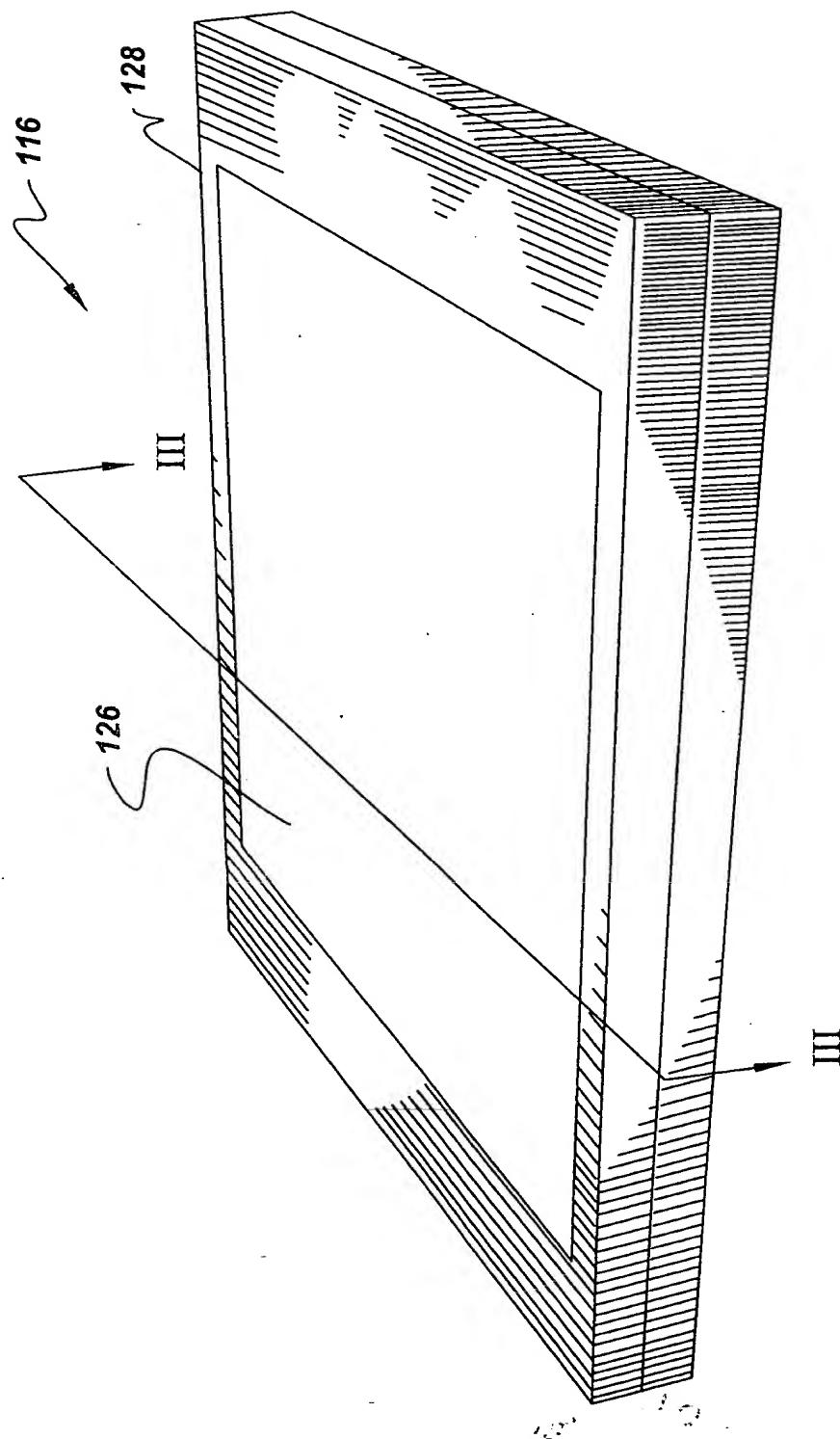


fig. 2
(PRIOR ART)

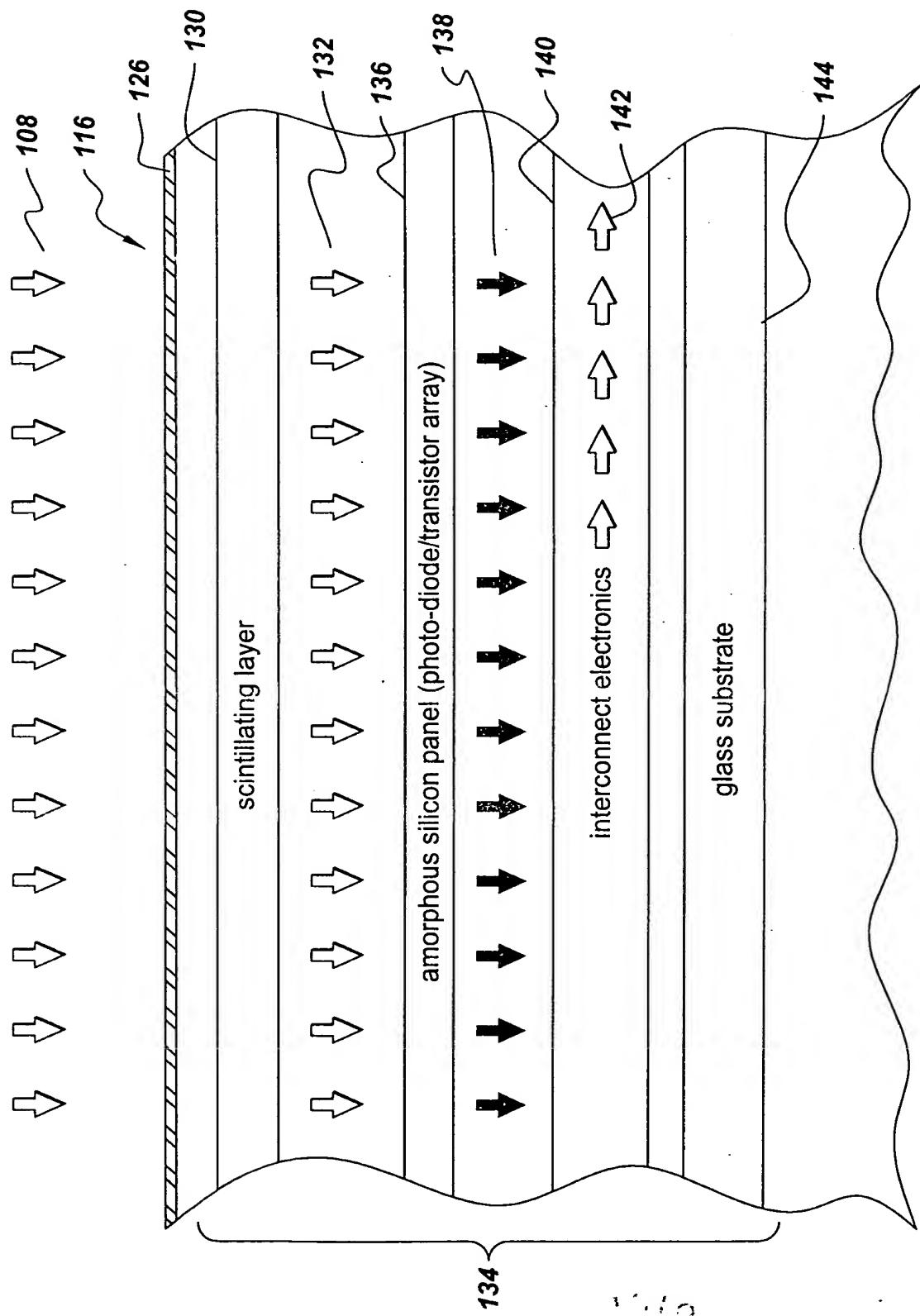
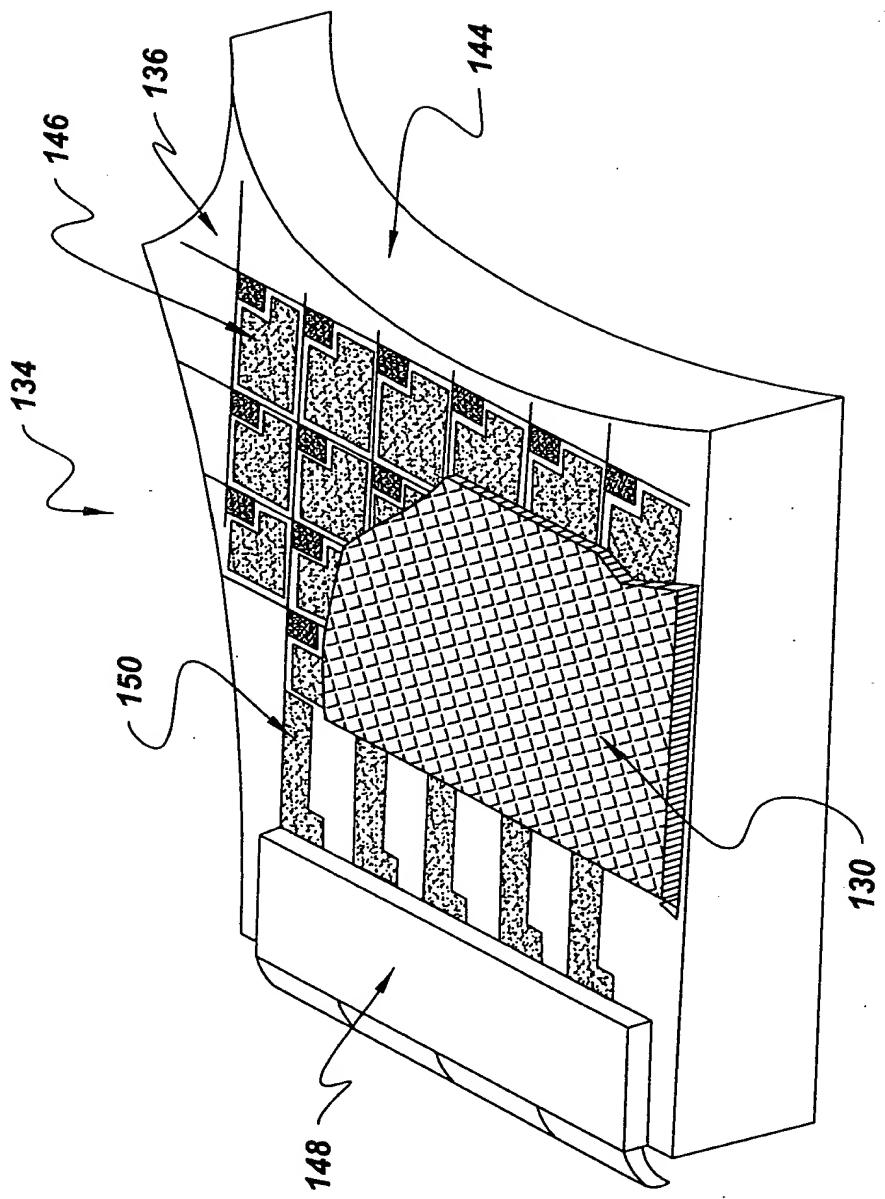


fig. 3

fig. 4
(Prior Art)



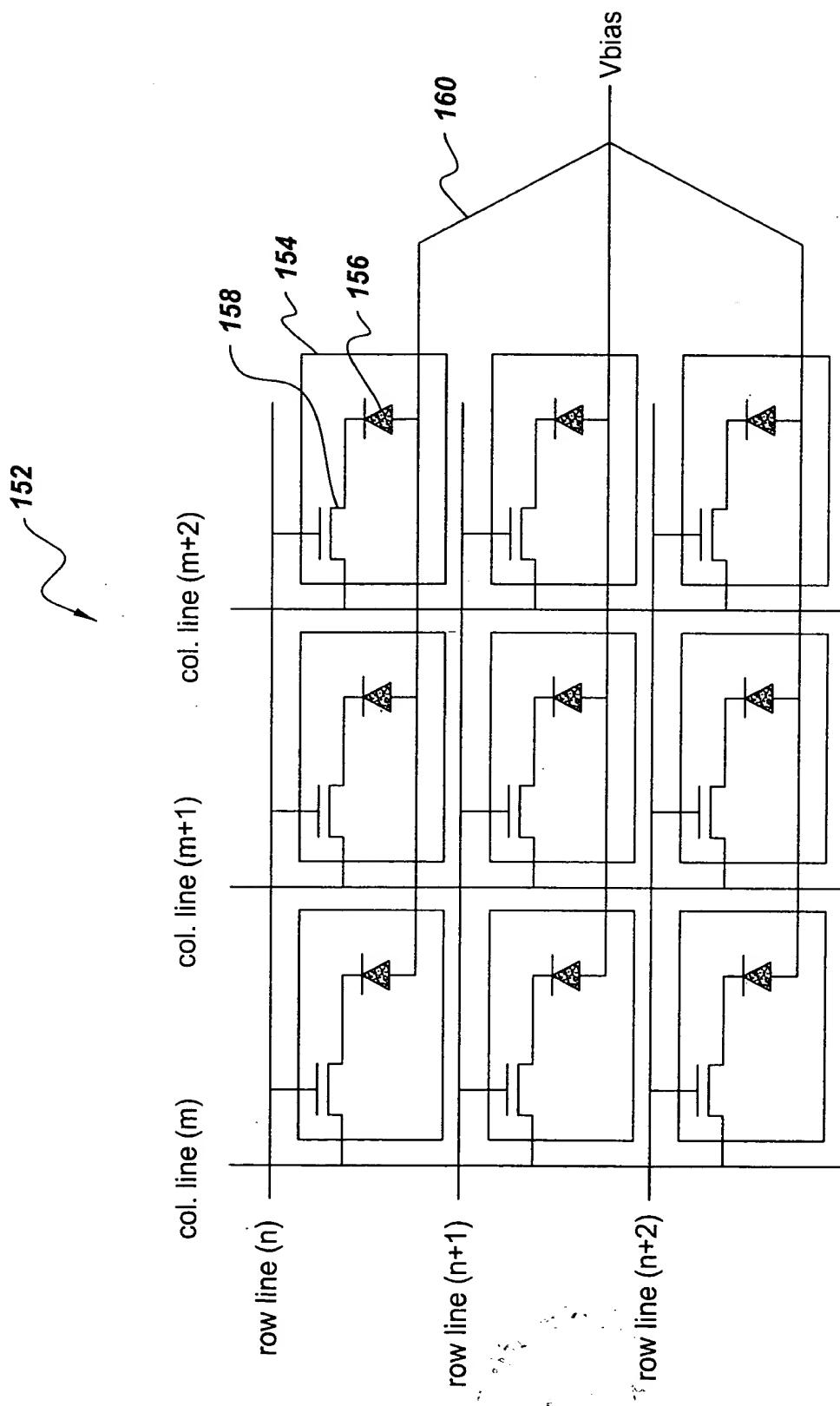


fig. 5
(Prior Art)

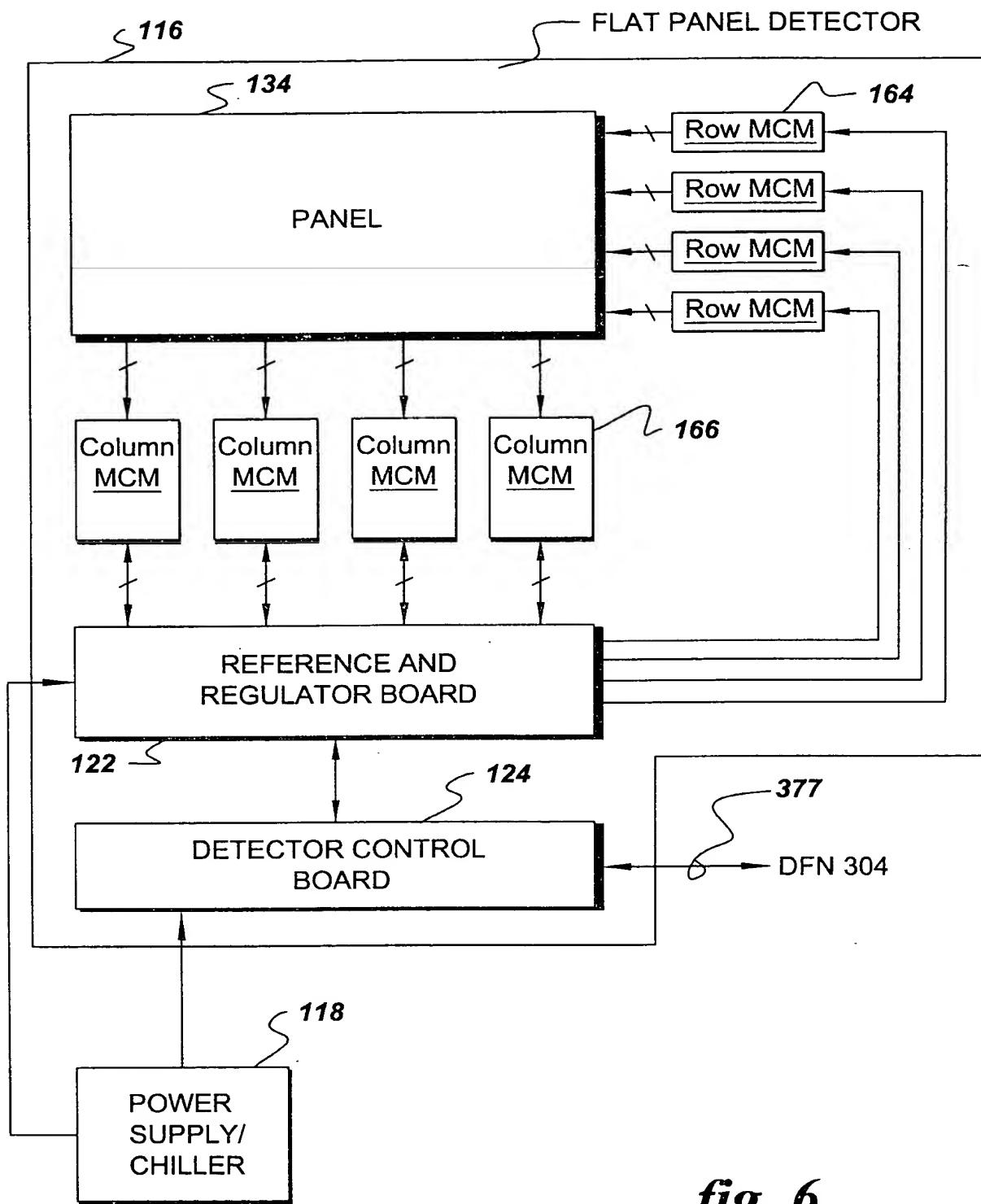


fig. 6
(PRIOR ART)

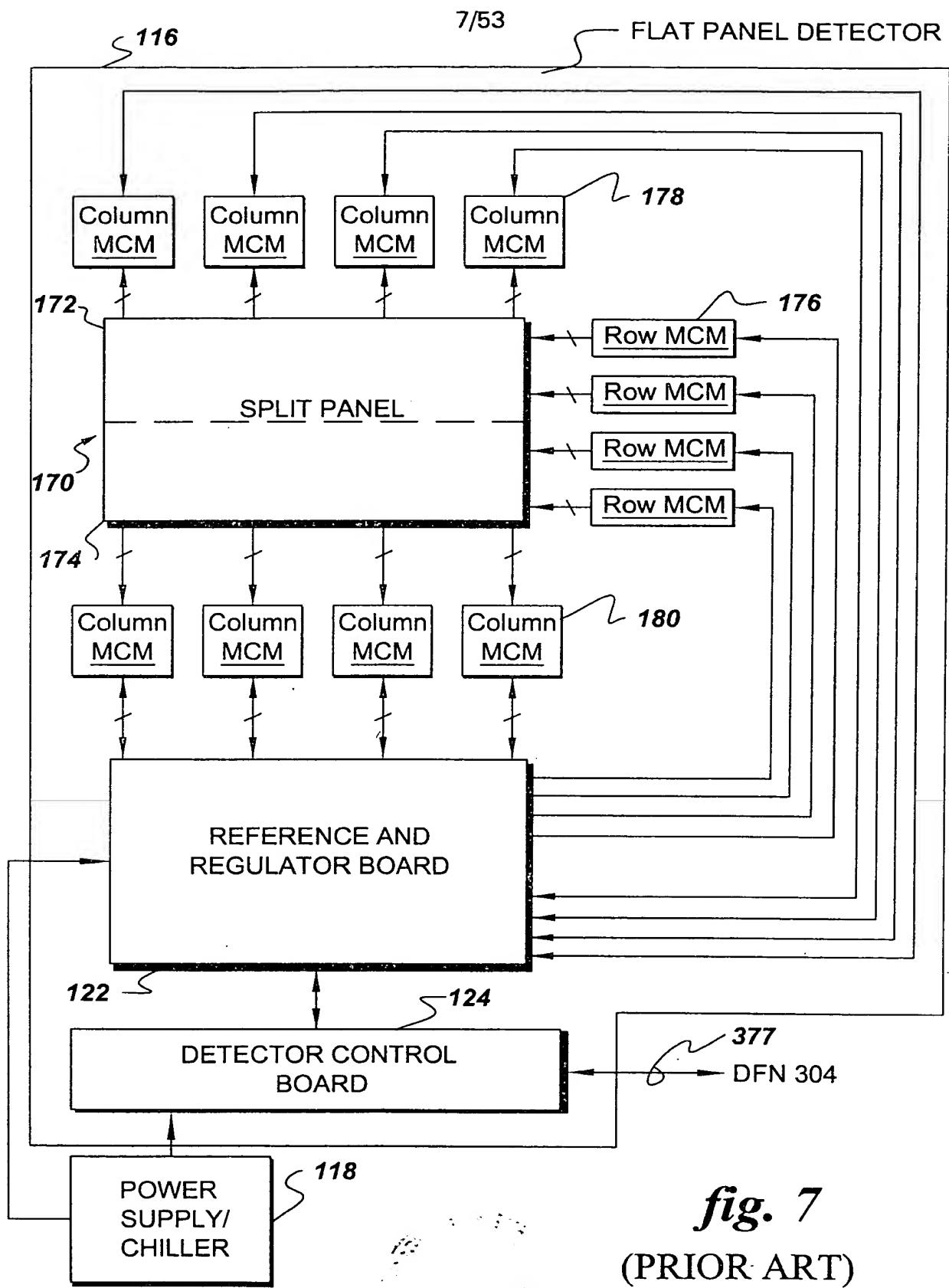
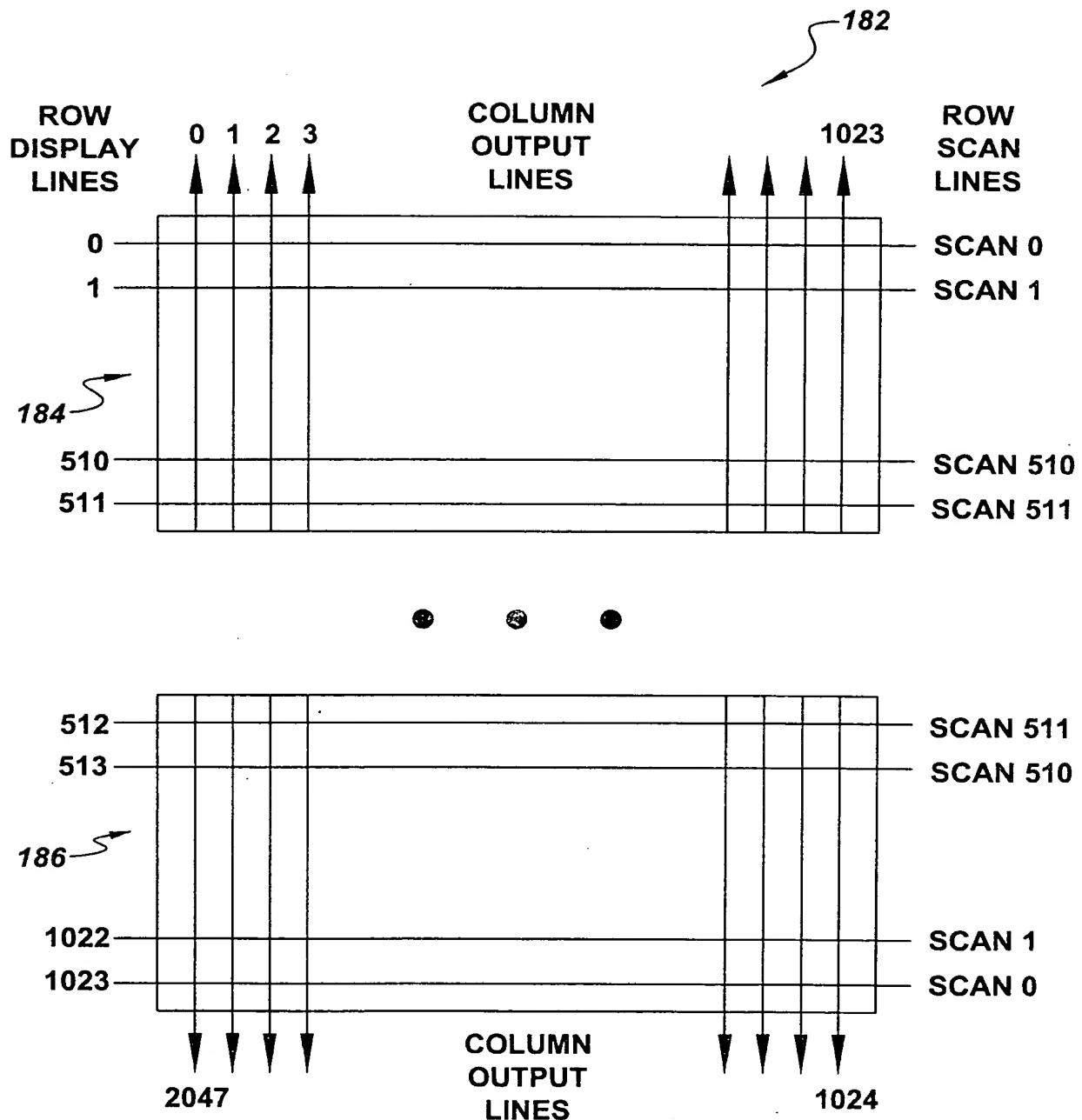


fig. 7
(PRIOR ART)



CARDIAC/SURGICAL DIGITAL X-RAY PANEL

fig. 8
(PRIOR ART)

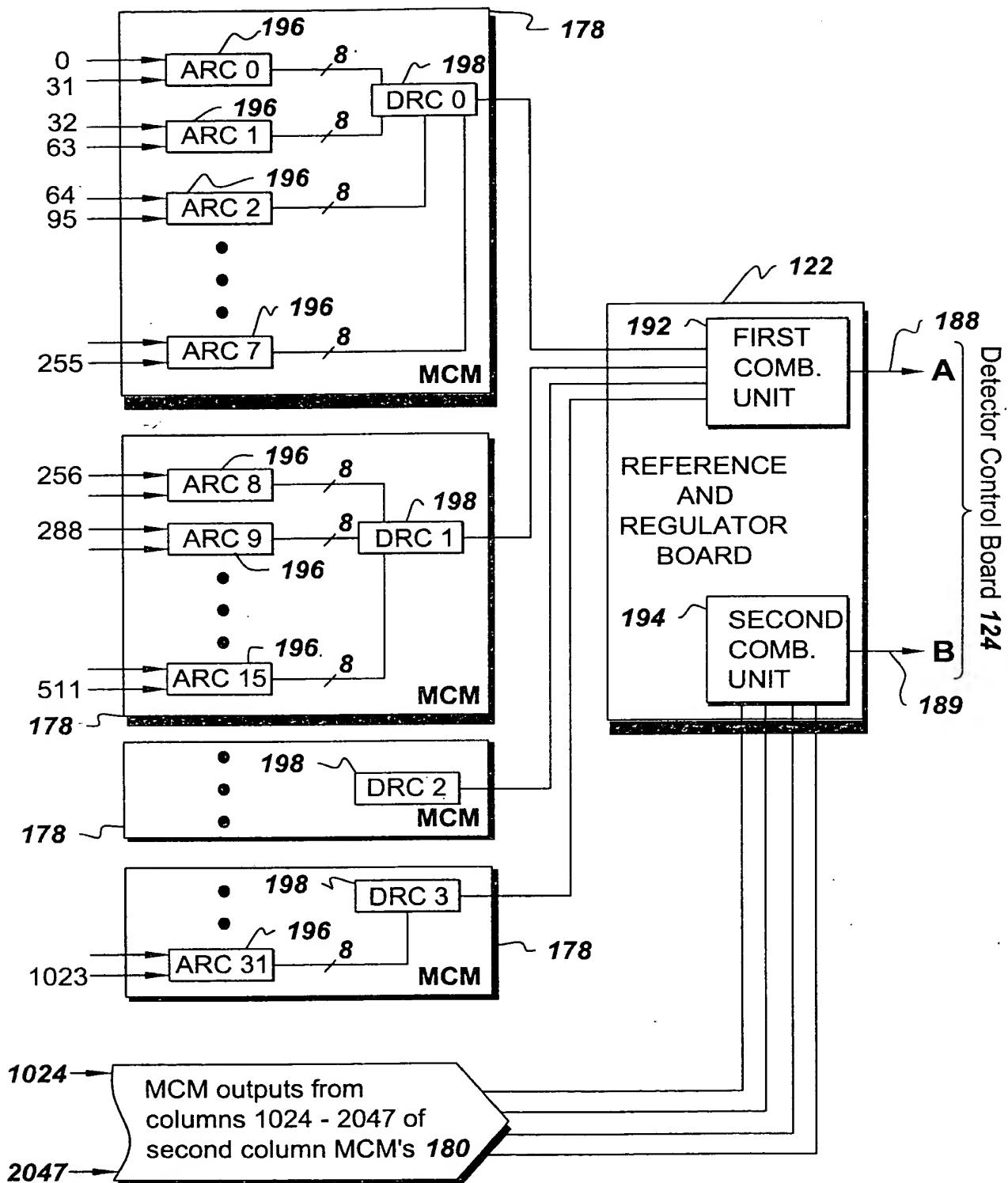
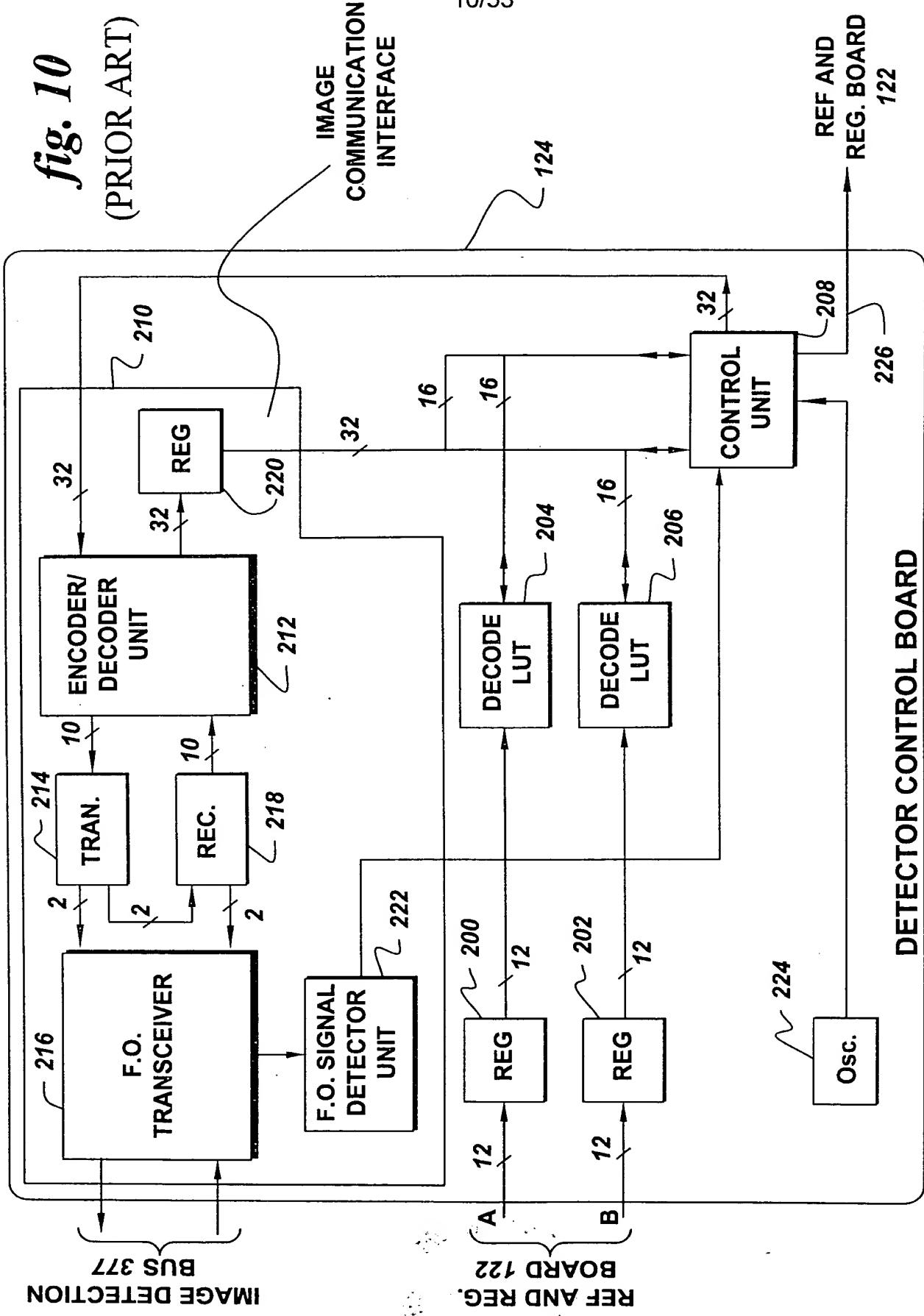
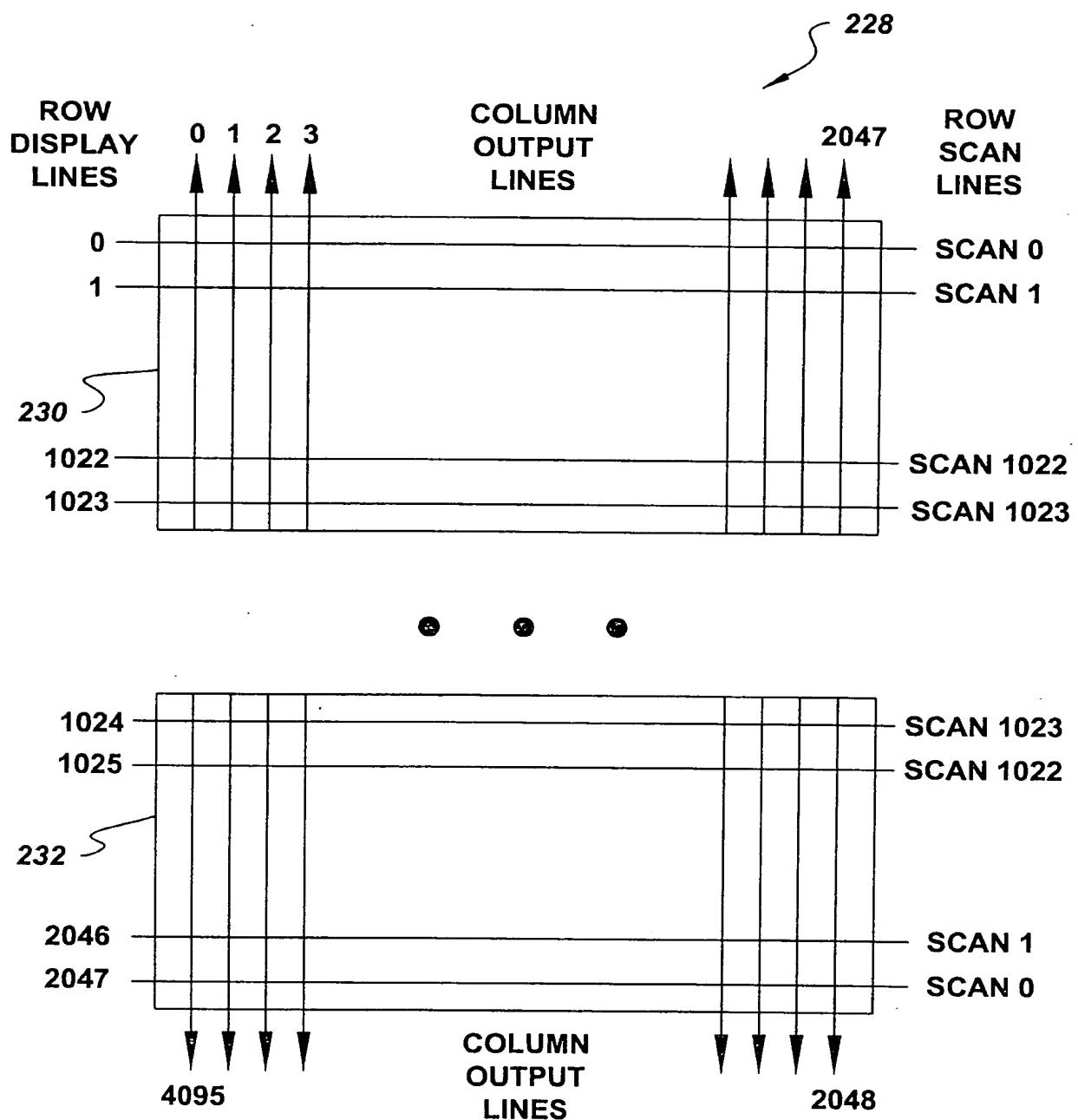


fig. 9 (PRIOR ART)

fig. 10
(PRIOR ART)





RADIOGRAPHY DIGITAL X-RAY PANEL

fig. 11 (PRIOR ART)

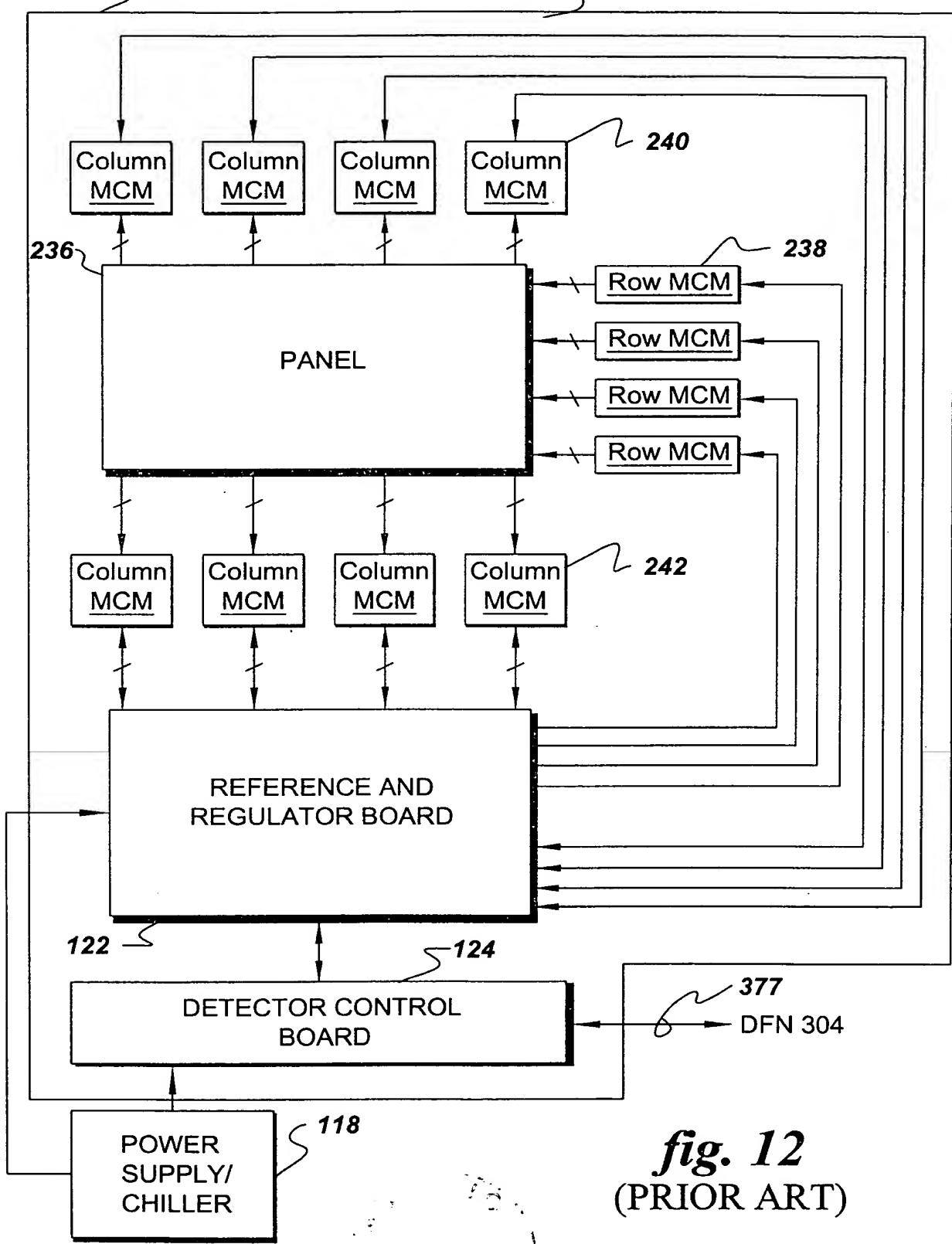
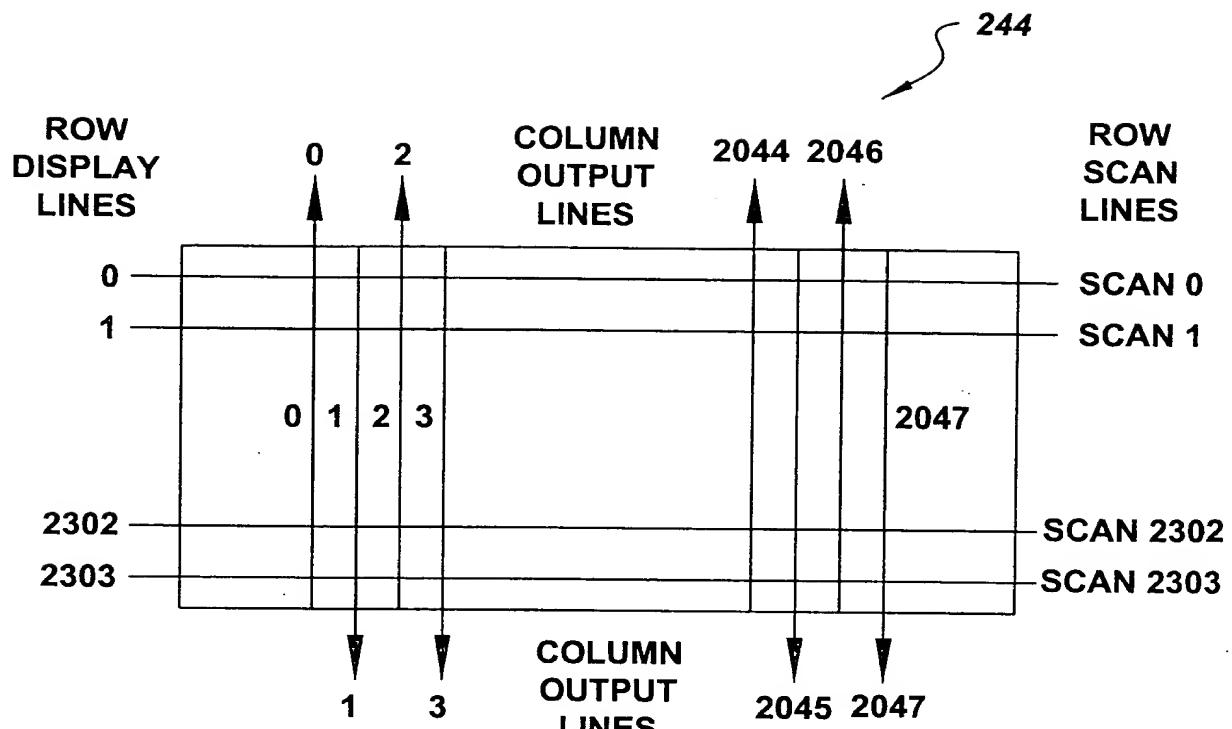


fig. 12
(PRIOR ART)



MAMMOGRAPHY DIGITAL X-RAY PANEL

fig. 13
(PRIOR ART)

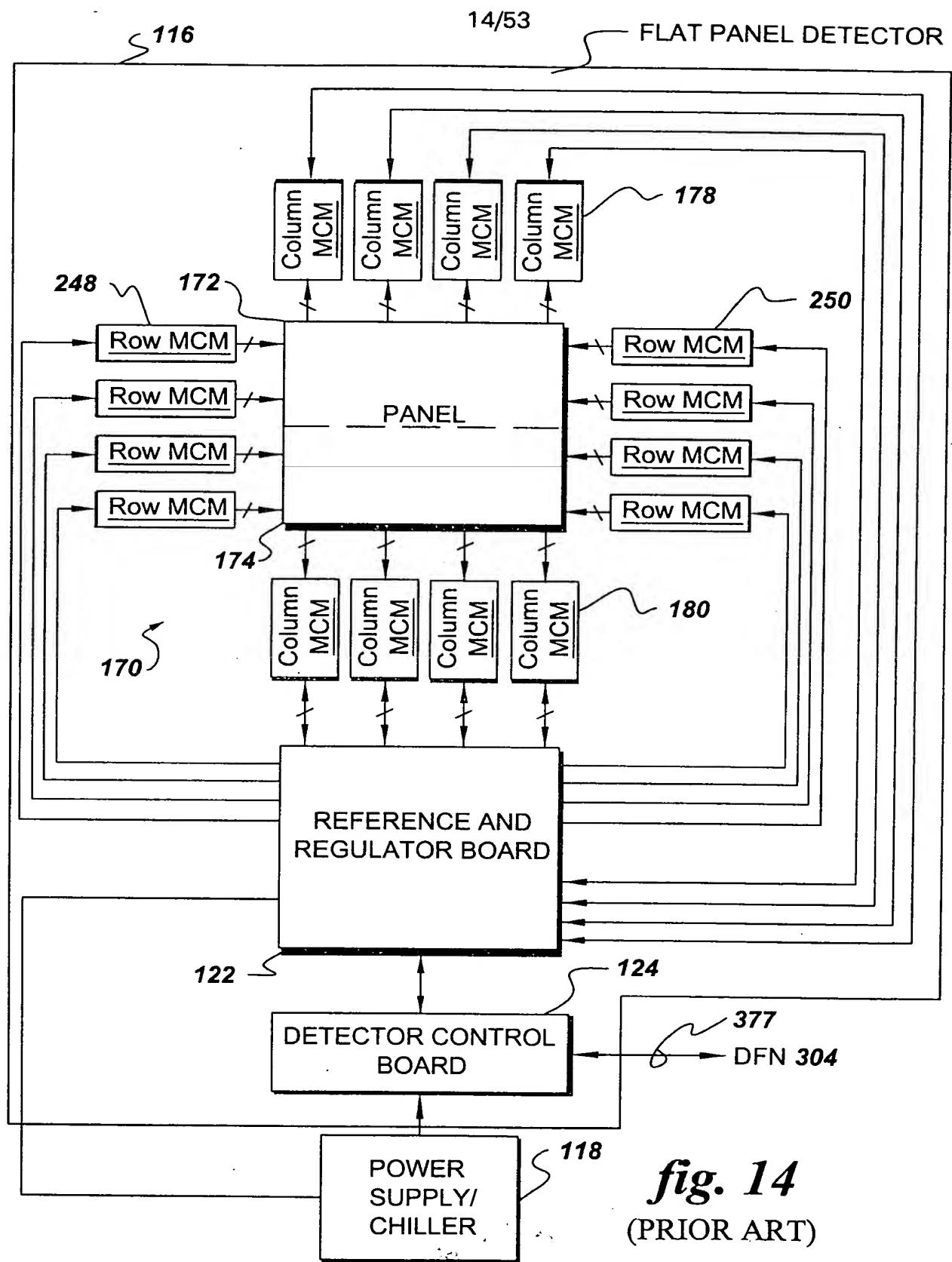


fig. 14
(PRIOR ART)

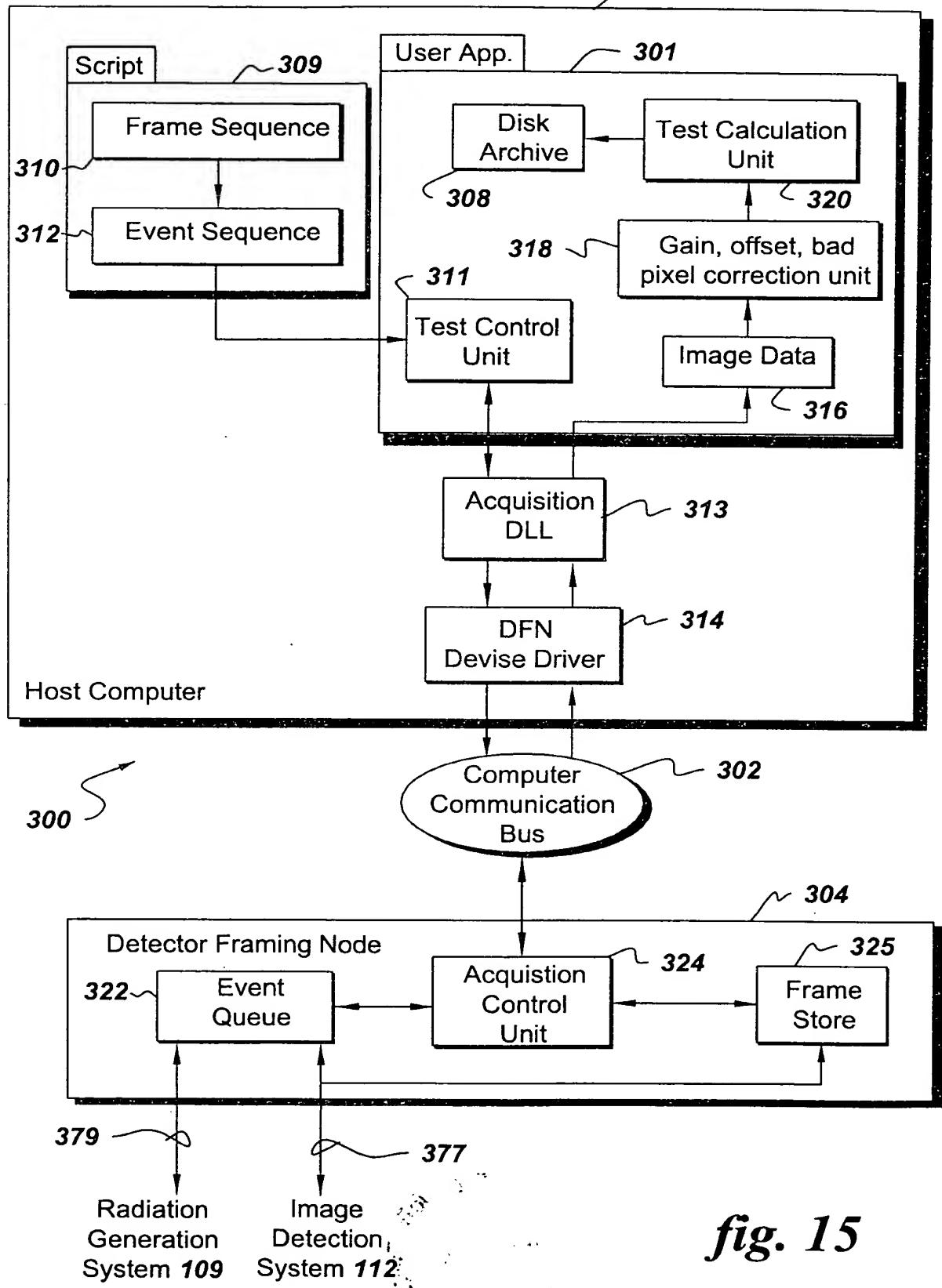
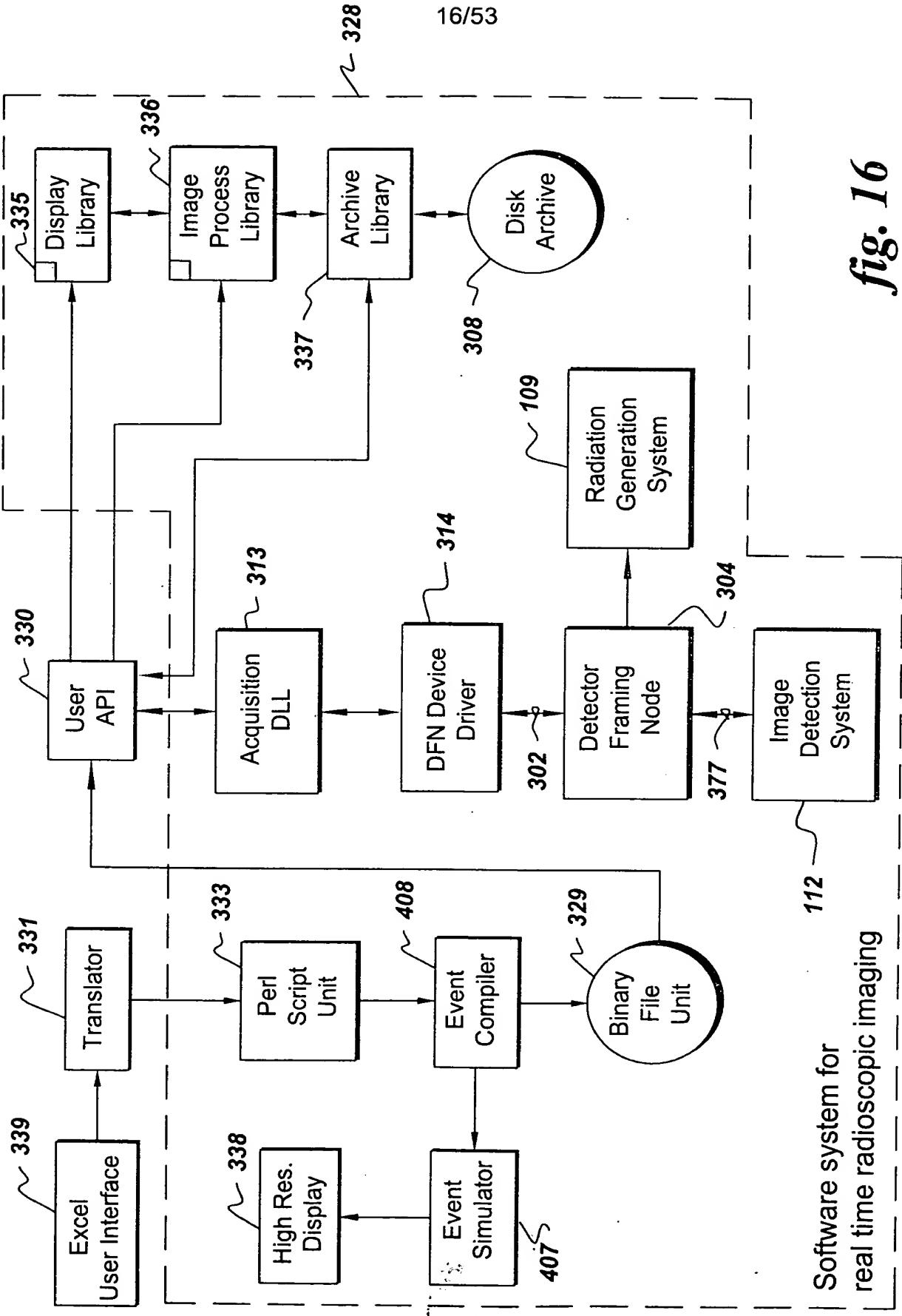
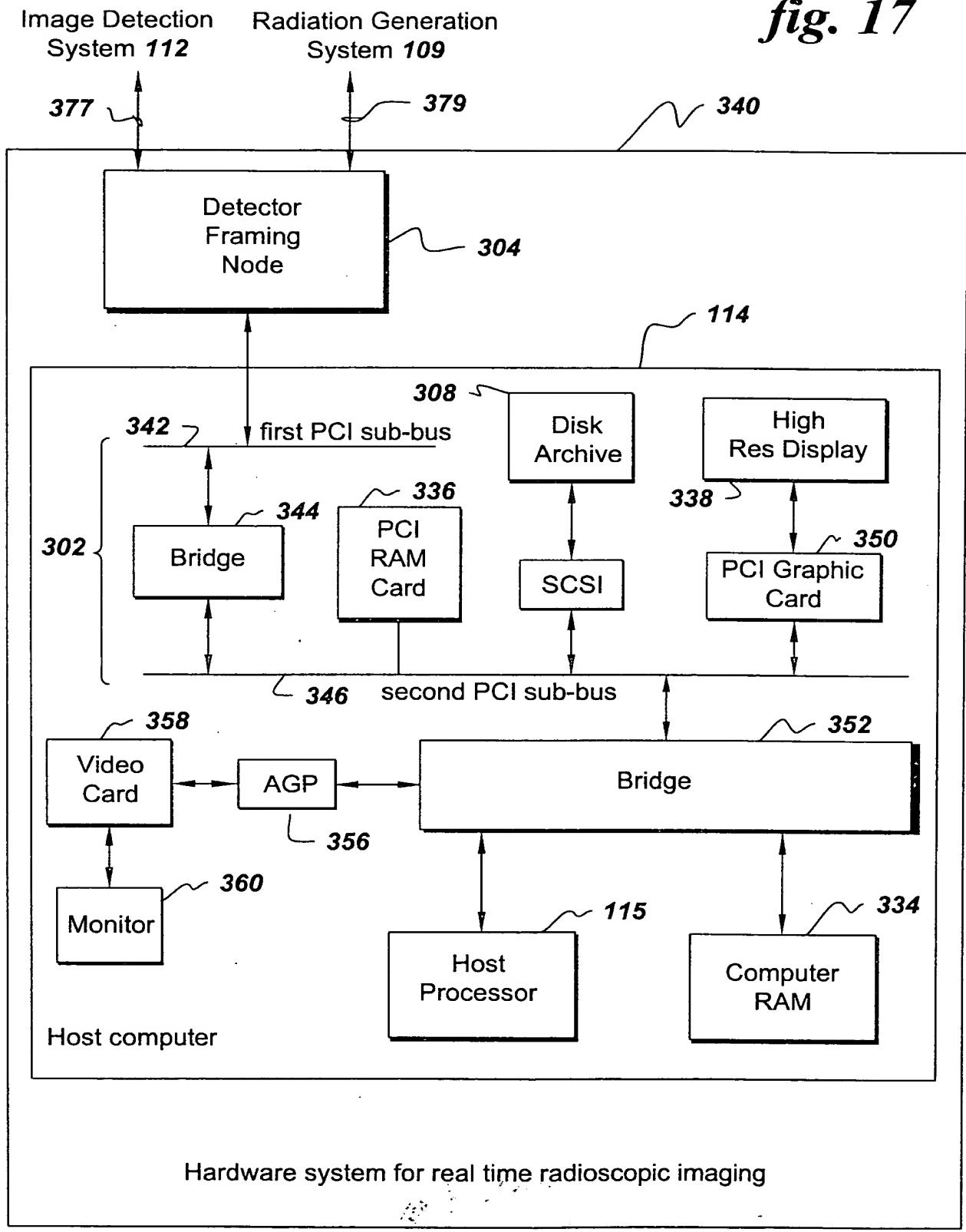


fig. 15



Software system for real time radioscopic imaging

fig. 17



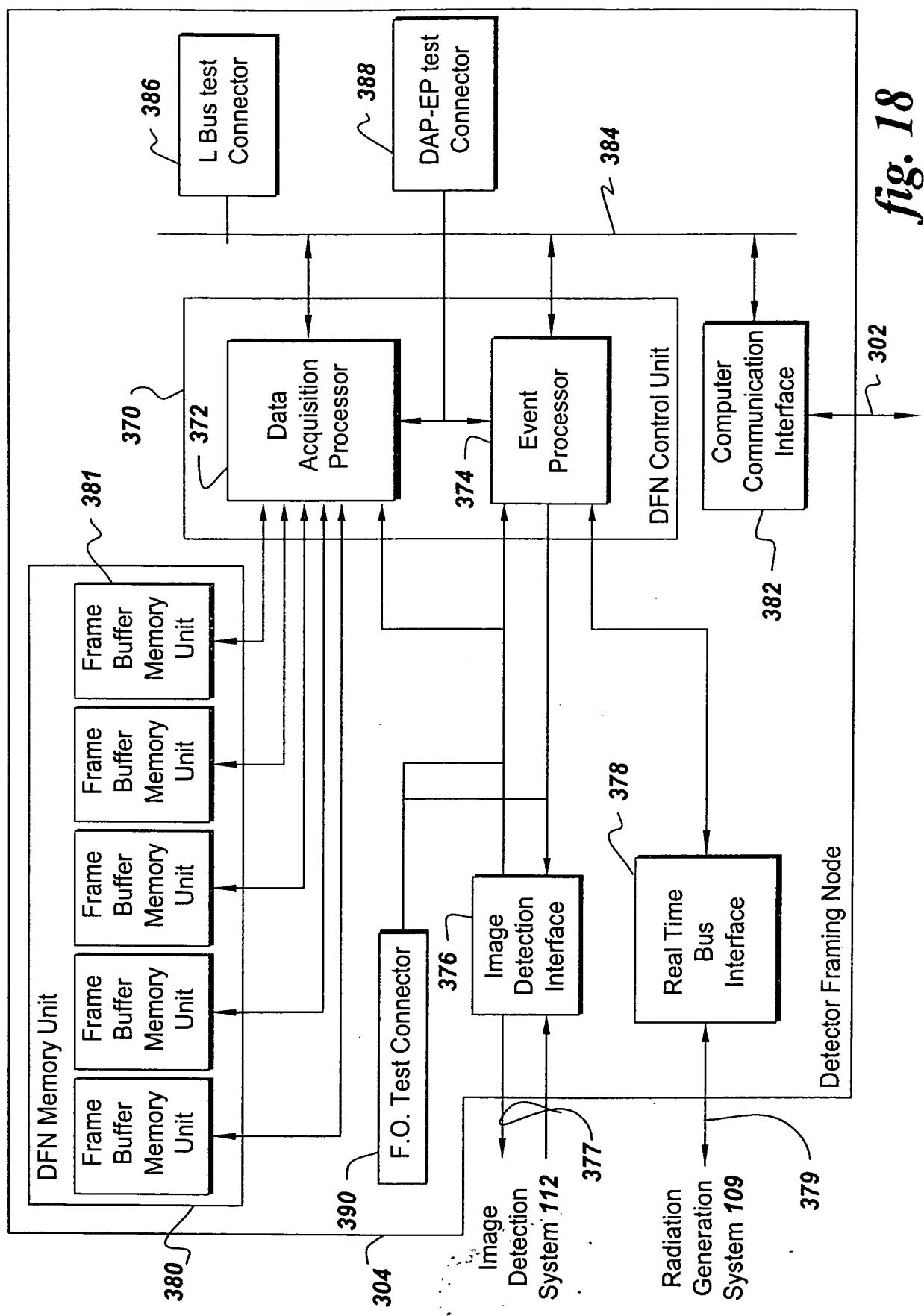


fig. 18

Host Computer 14

			(fm/sec)	length	latency	memory offset	gbr
Panel Setup	Real Time	30	unlimited	<5 frames	host	none	
Single Frame	Post Process	-	-	Delay ~.1 sec	"	y	
Single Frame	Post Process	-	-	Delay ~.2 sec	"	y	y
Real Time	Real Time	R	unlimited	<5 frames	host	none	
Real Time	Real Time	R-X	unlimited	<5 frames	"	y	
Real Time	Real Time	R-Y	unlimited	<5 frames	"	y	y

fig. 19

Modality	image size	Frames Stored host memory
Cardiac	1024 X 1024	200
Rad	2048 X 2048	50
Mammo	2304 X 2048	44

fig. 20

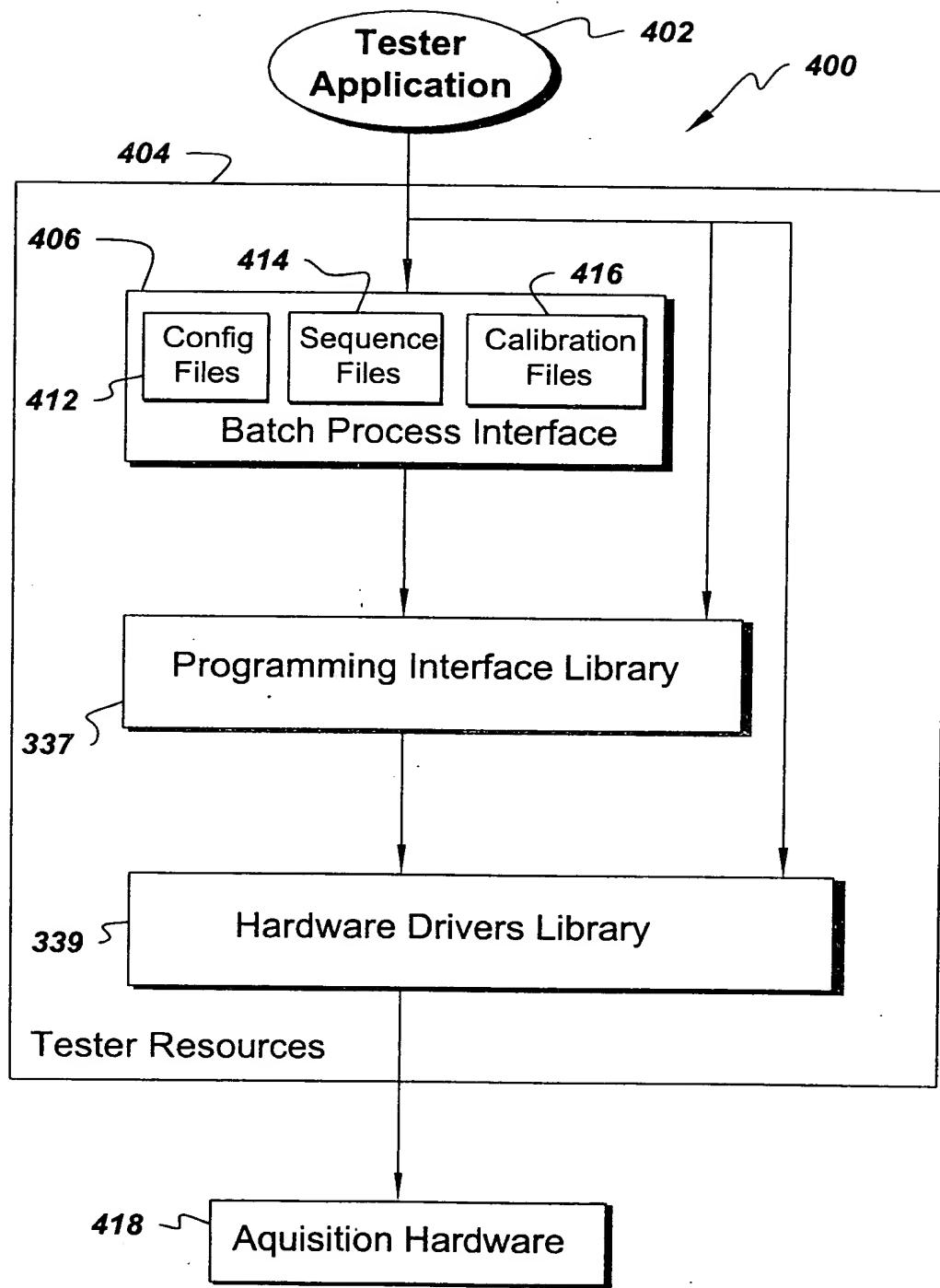


fig. 21

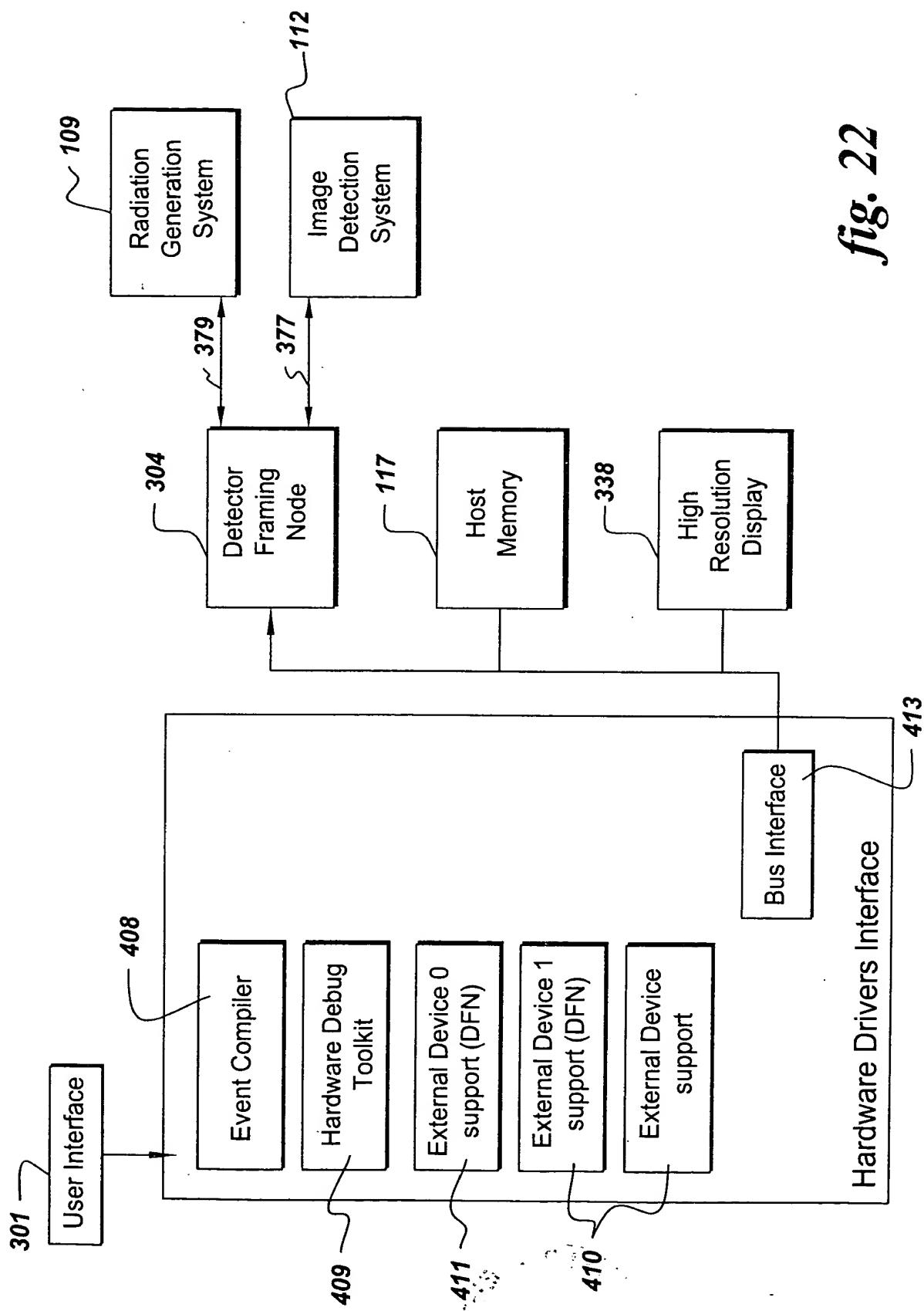


fig. 22

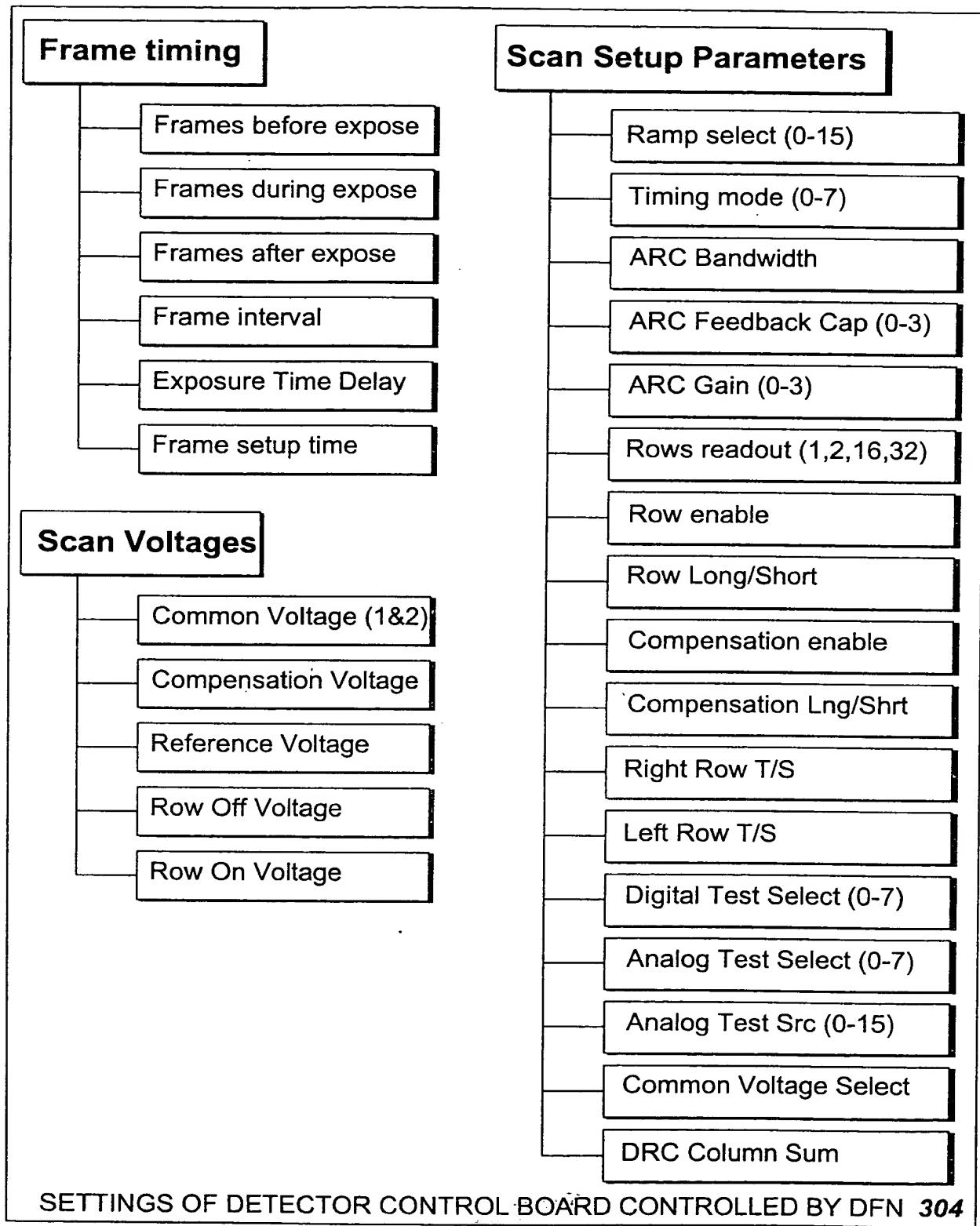
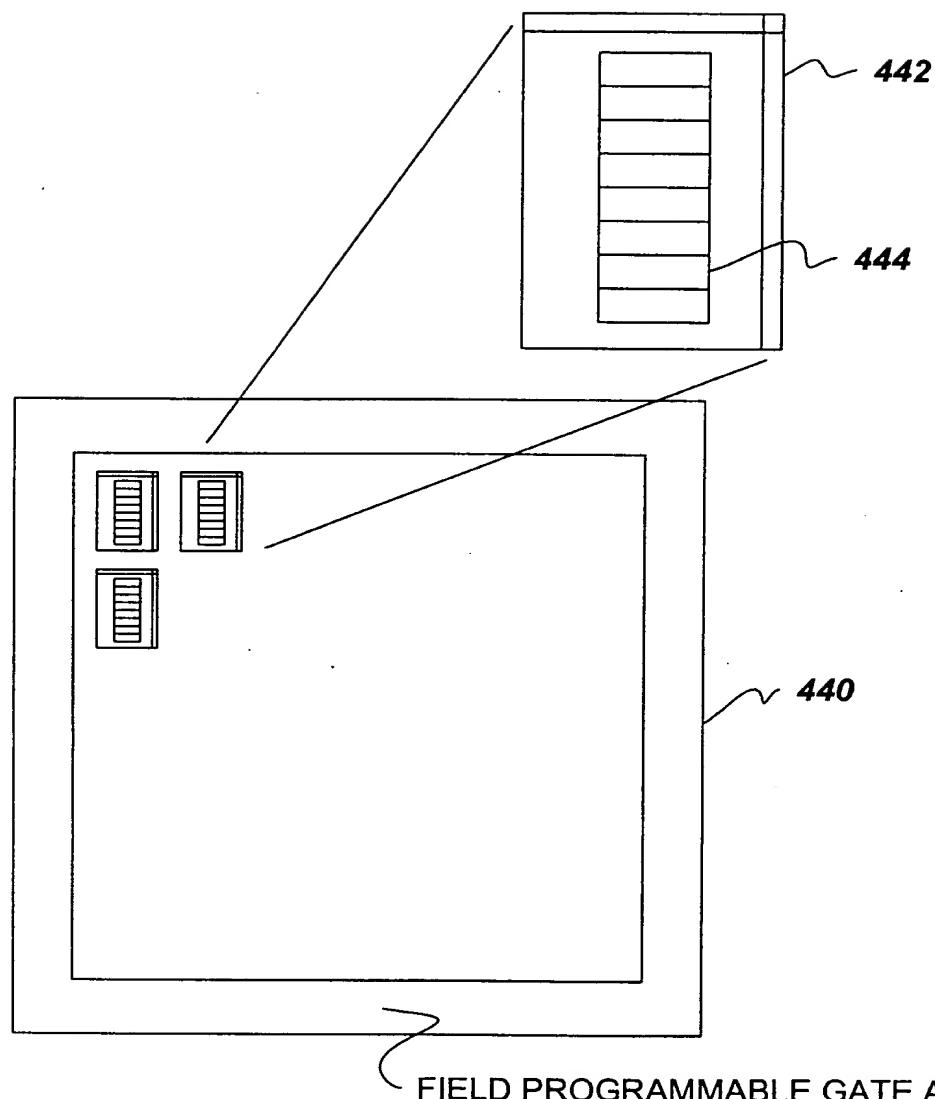


fig. 23



FIELD PROGRAMMABLE GATE ARRAY

fig. 24

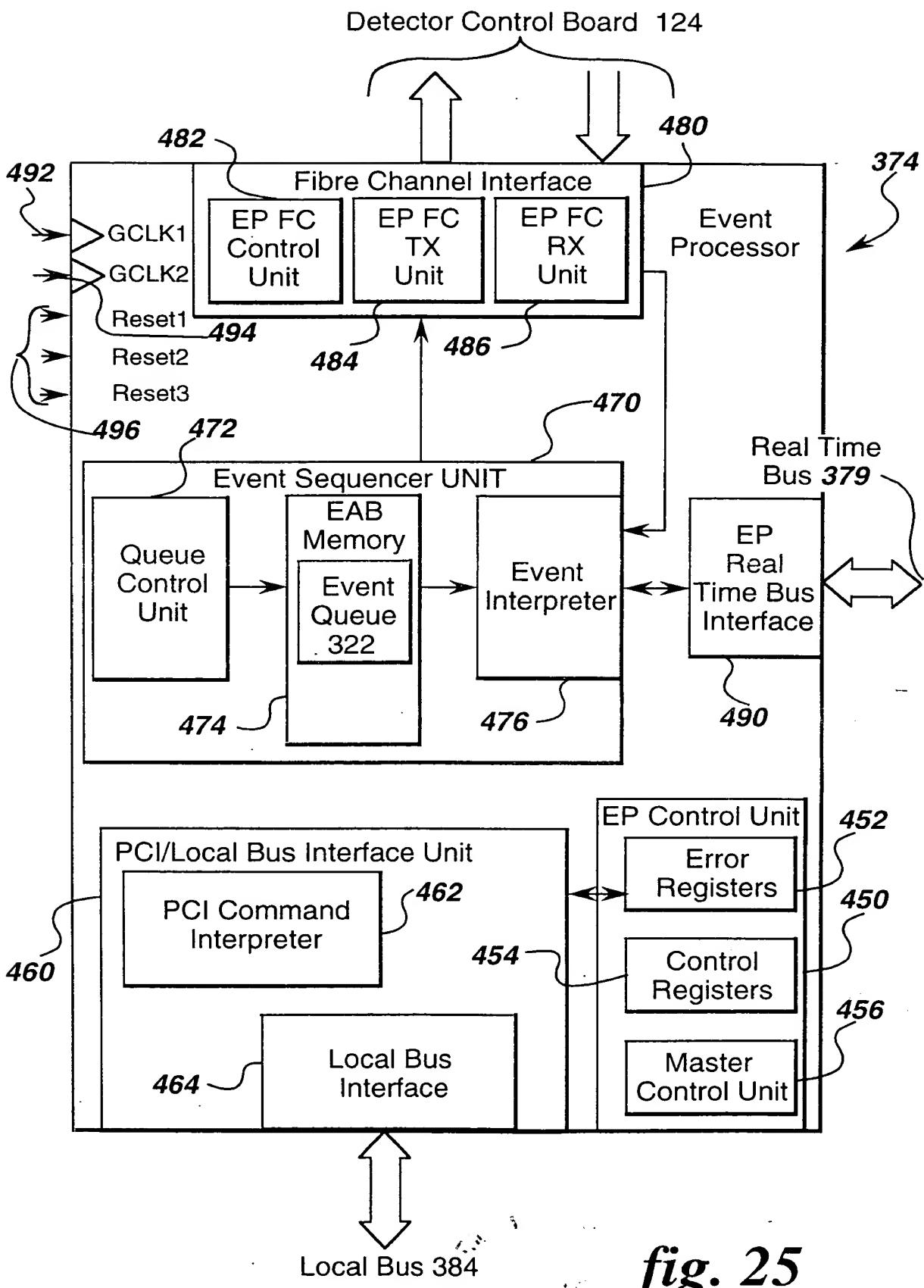
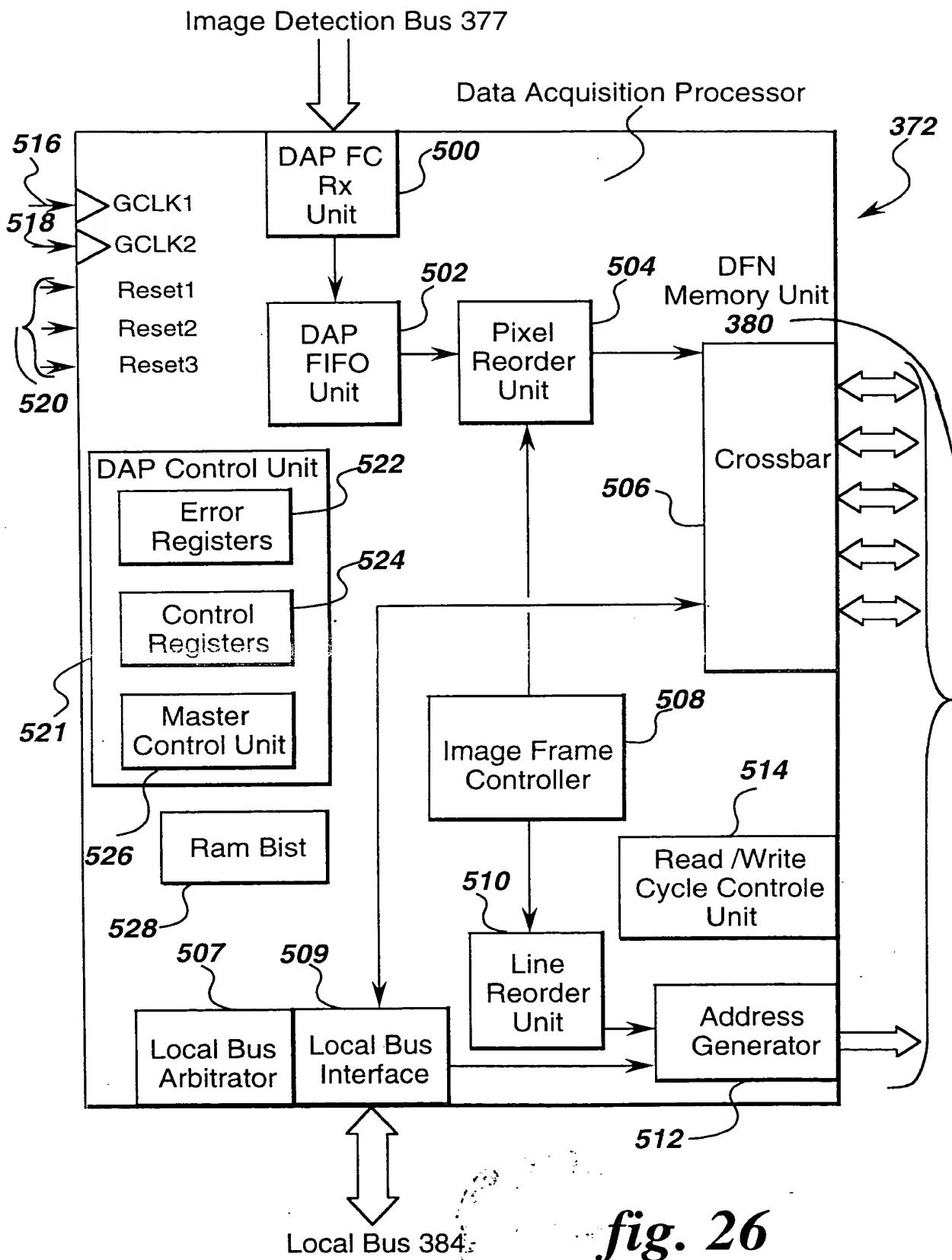


fig. 25



Local Bus 384 fig. 26

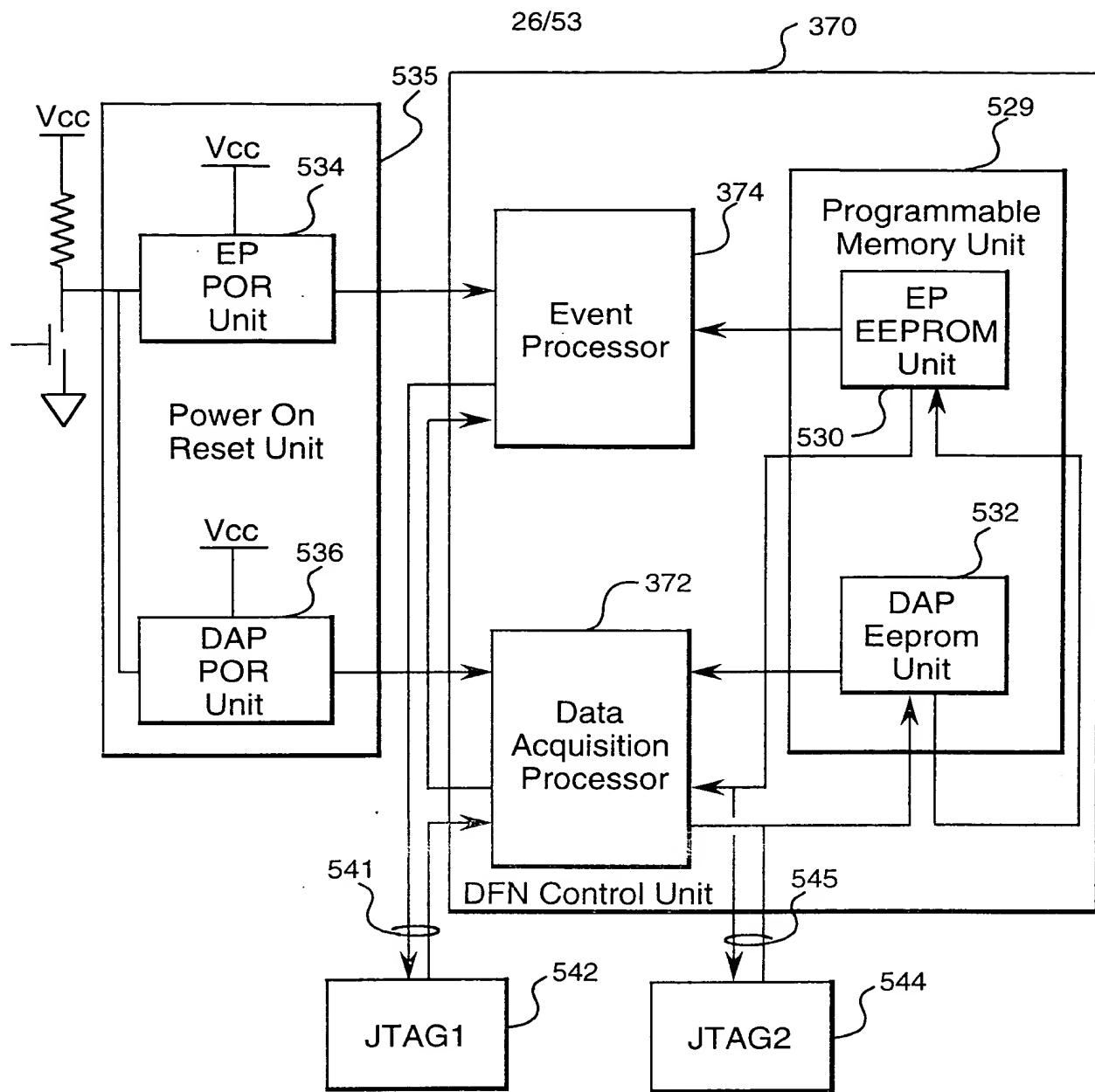


fig. 27

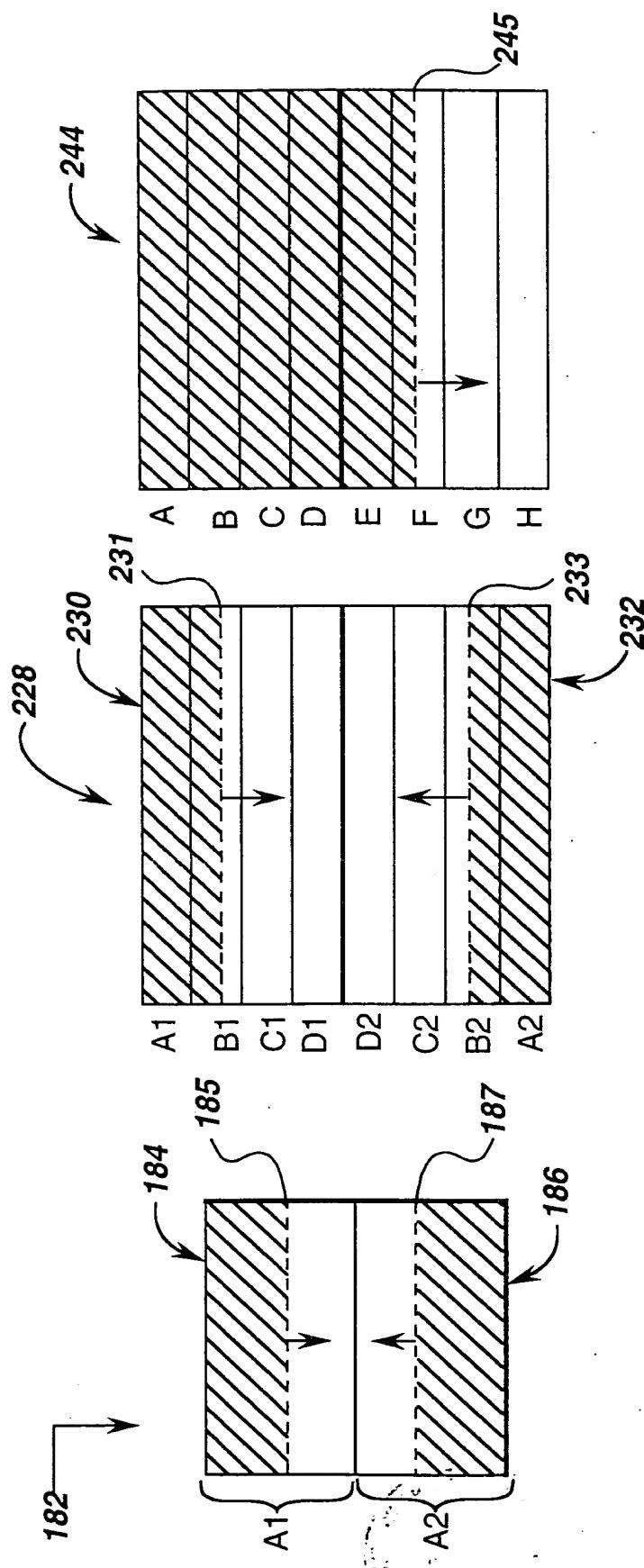


fig. 28

fig. 29

fig. 30

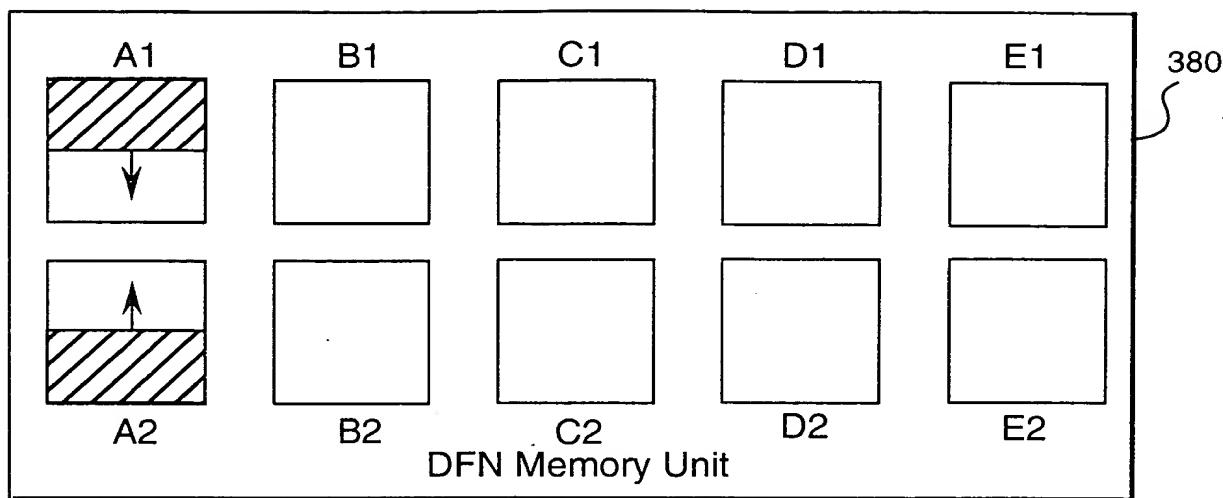


fig. 31

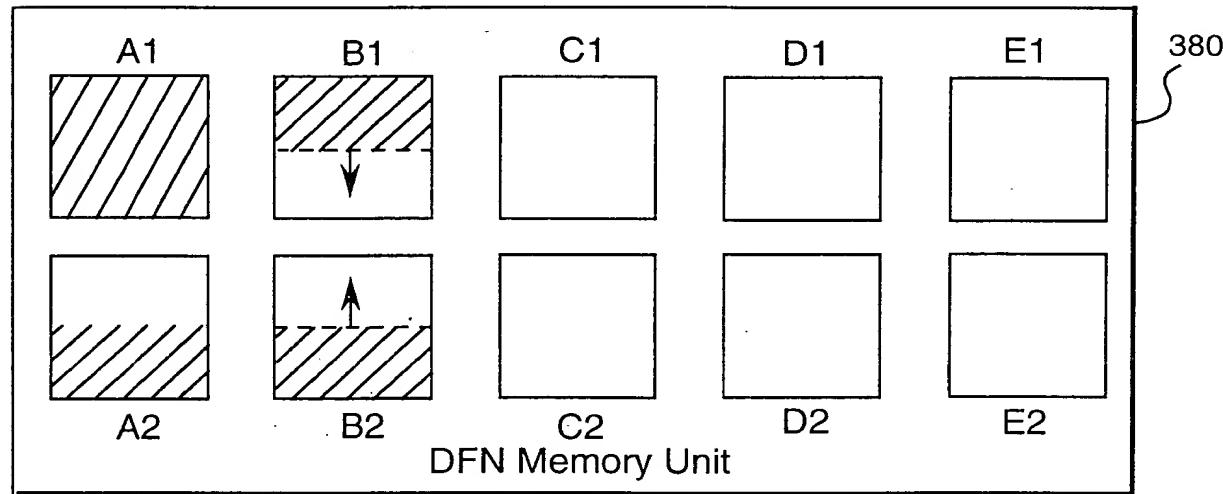


fig. 32

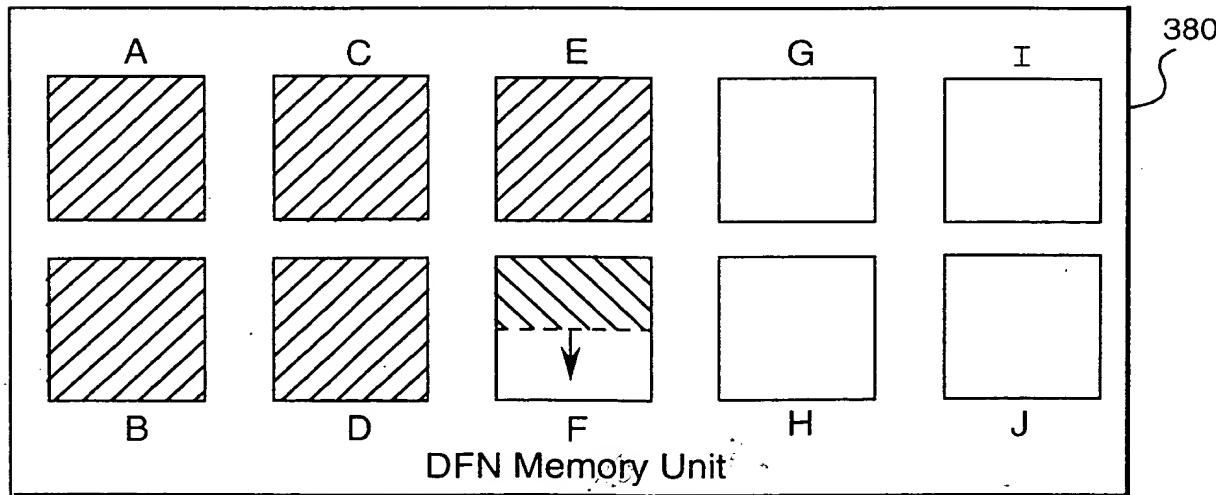


fig. 33

fig. 34

A1	A2
----	----

334

fig. 35

A1	B1	C1	D1	D2	C2	B2	A2
----	----	----	----	----	----	----	----

334

fig. 36

A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---

334

384 382 302

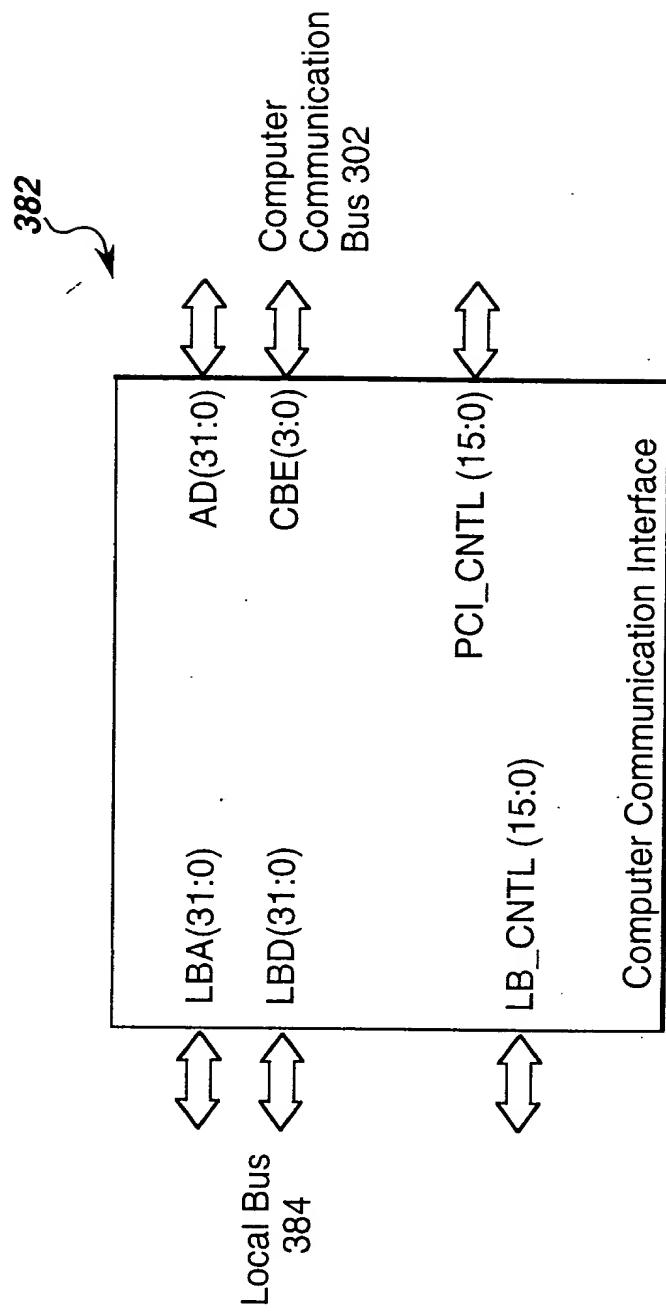


fig. 37

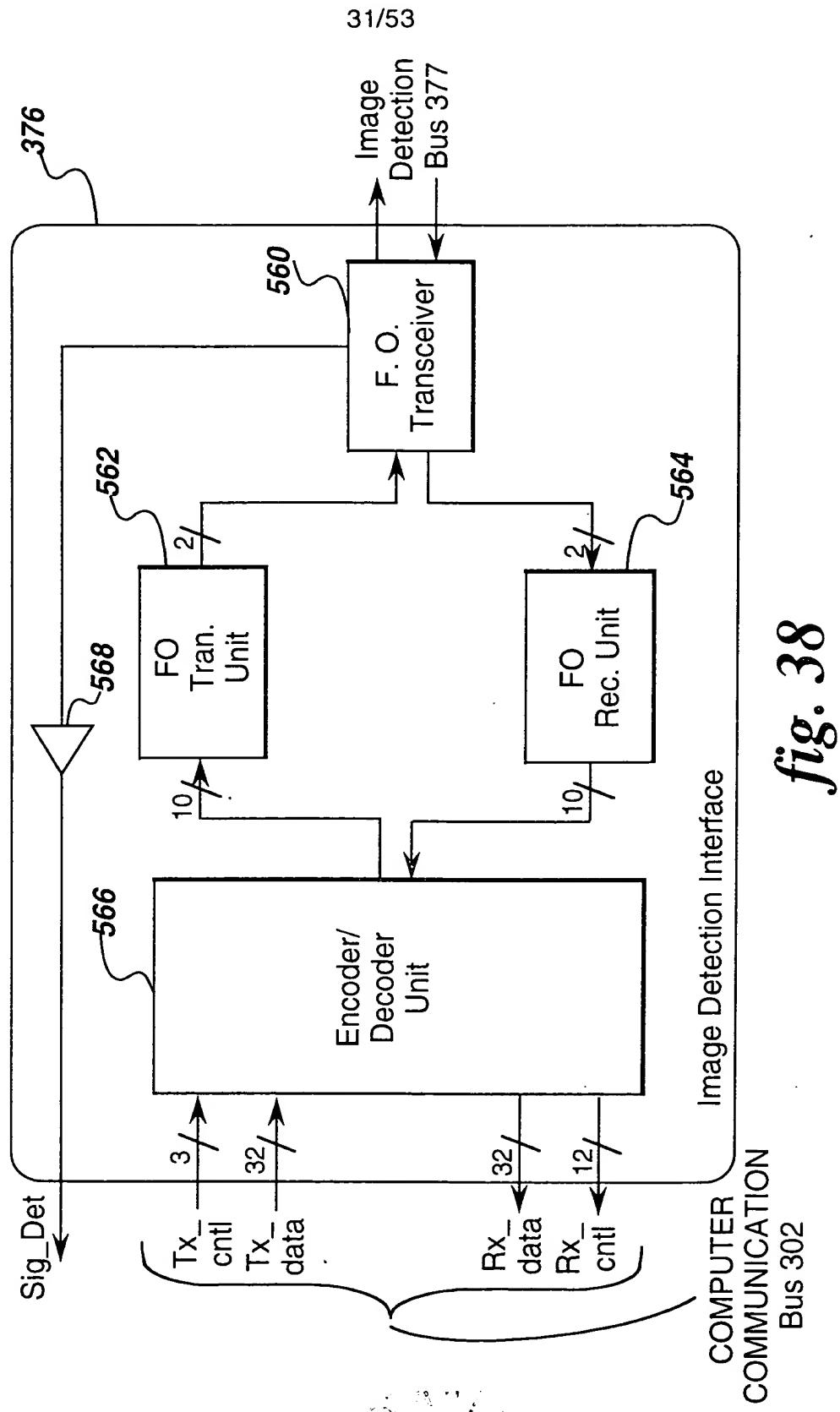


fig. 38

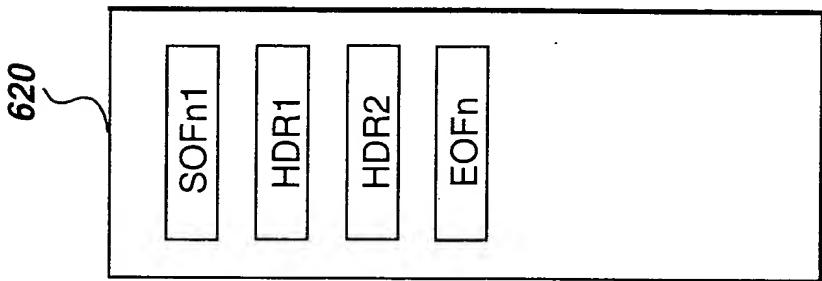
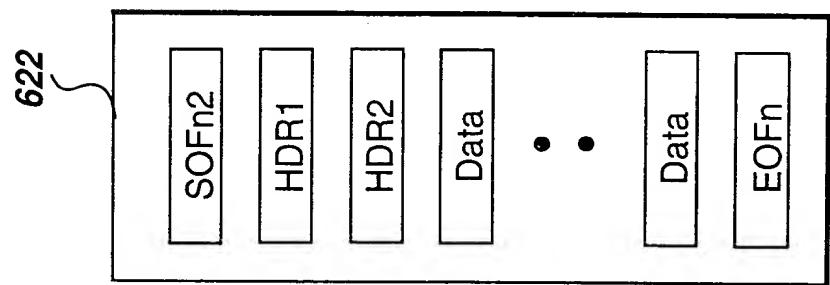
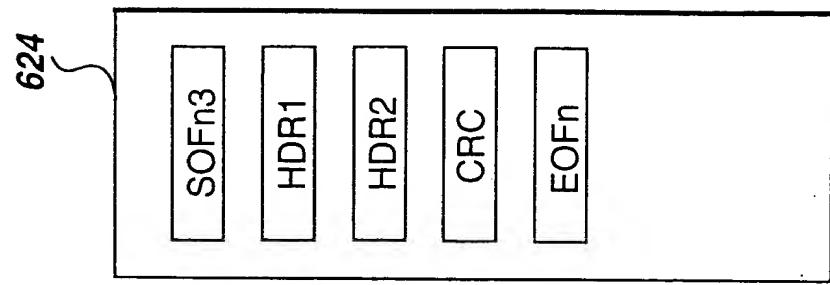


fig. 39

fig. 40

fig. 41

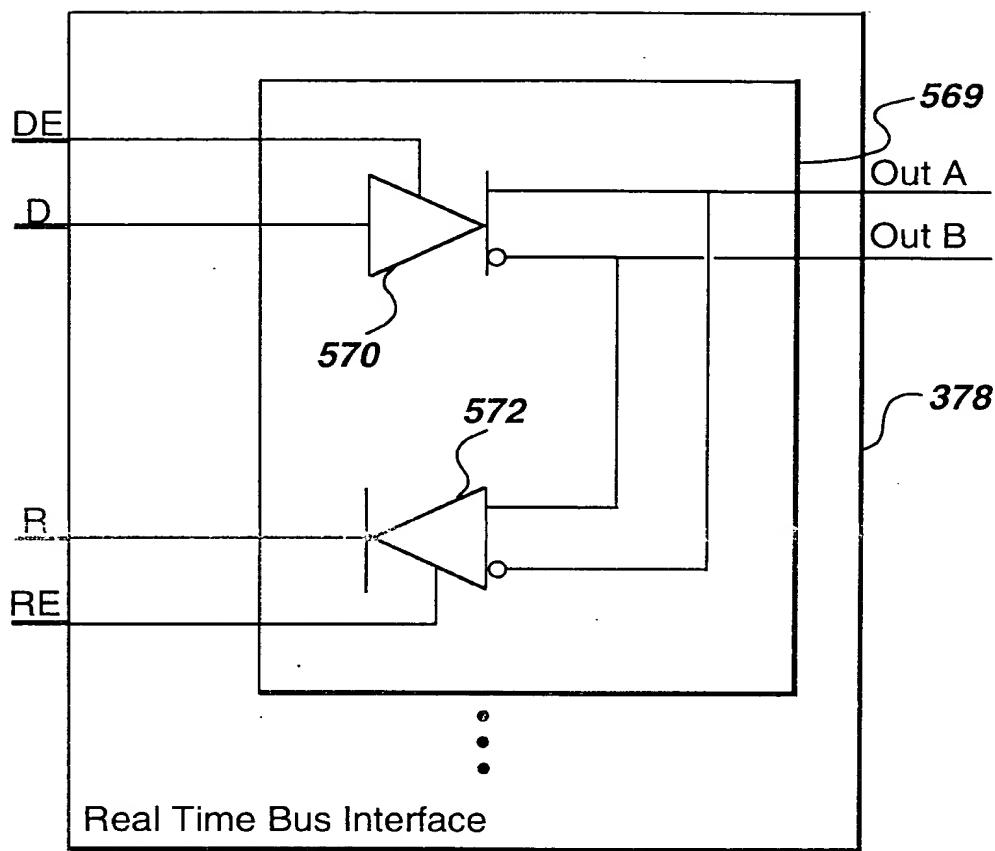
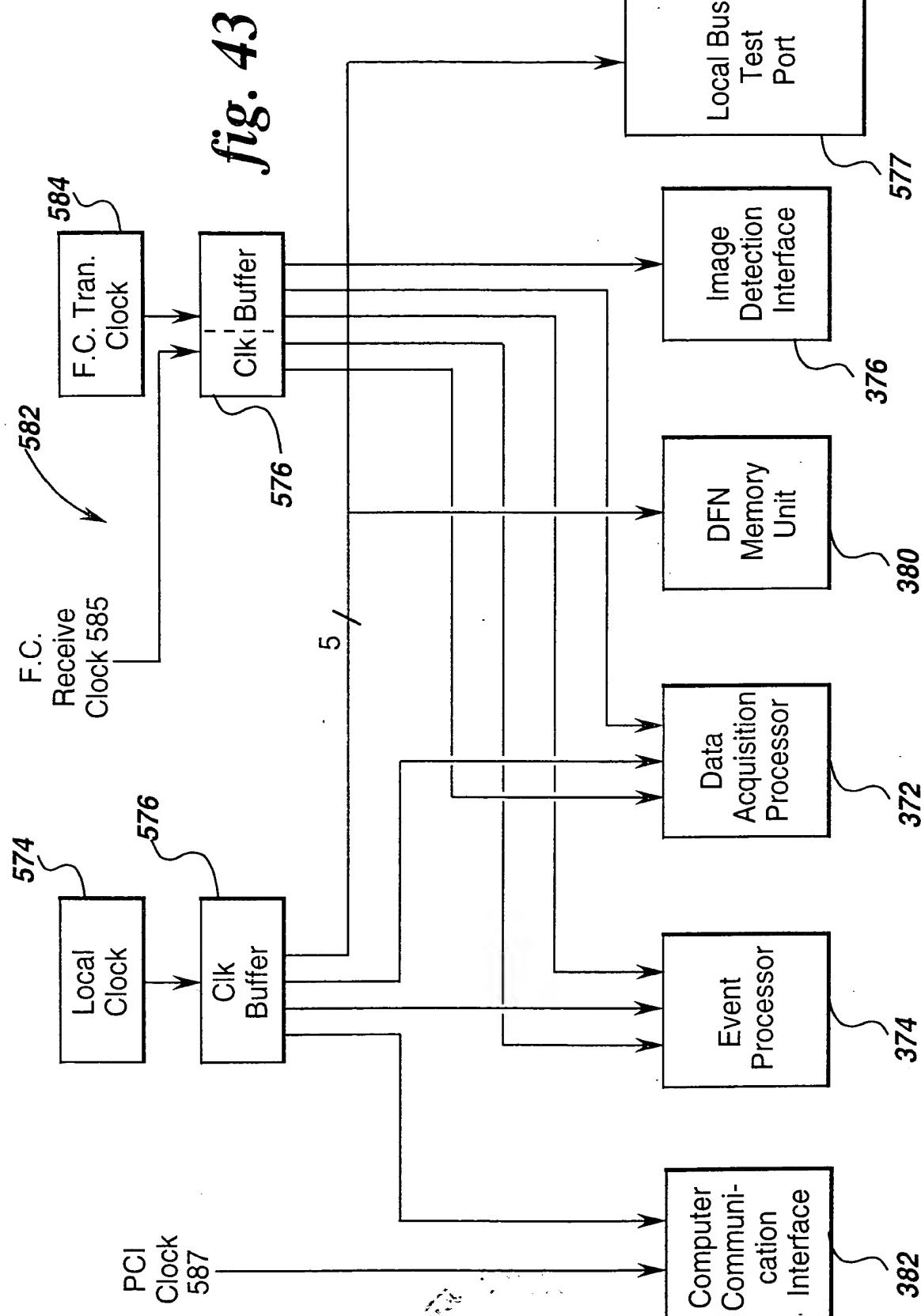
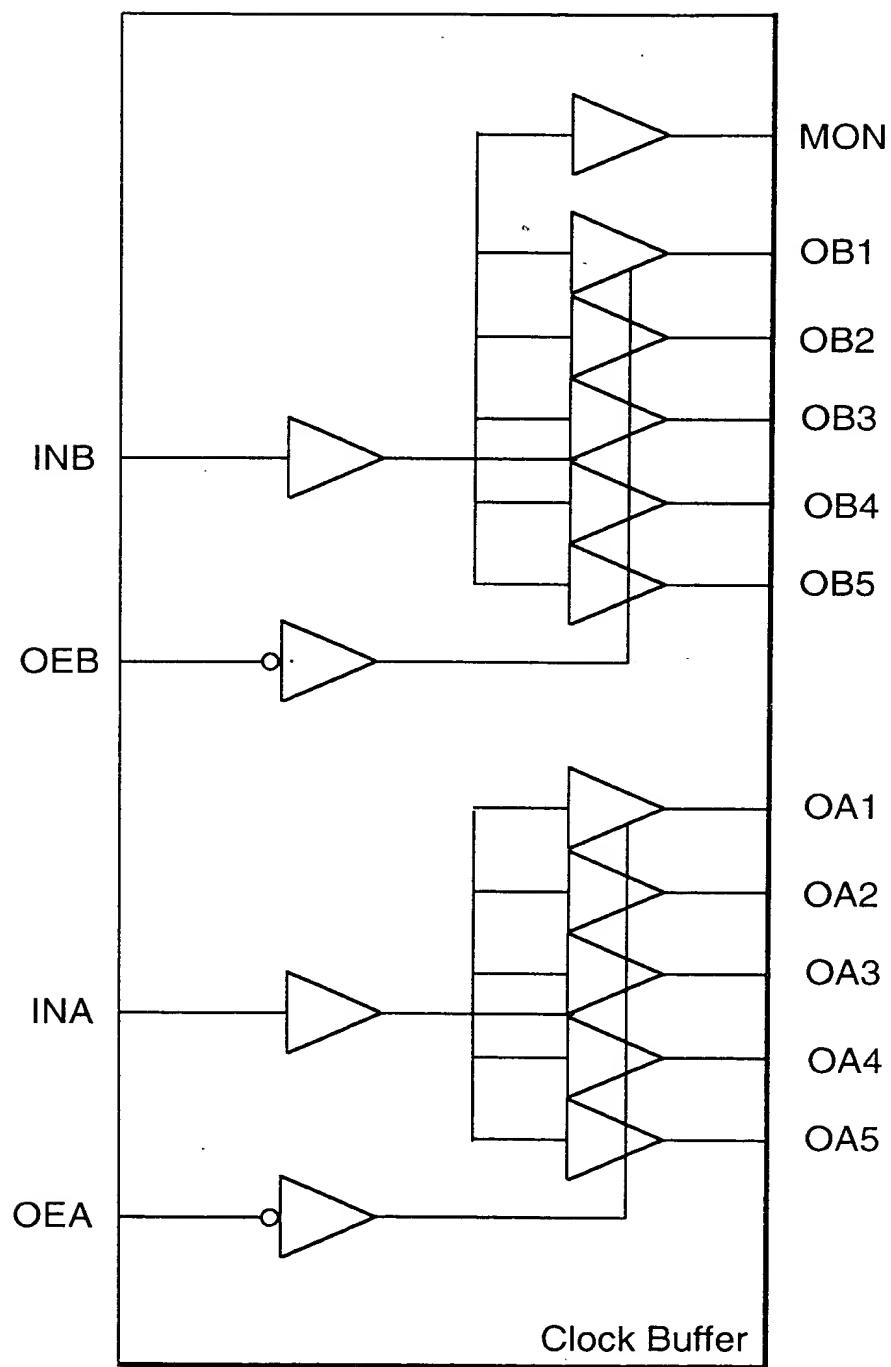


fig. 42





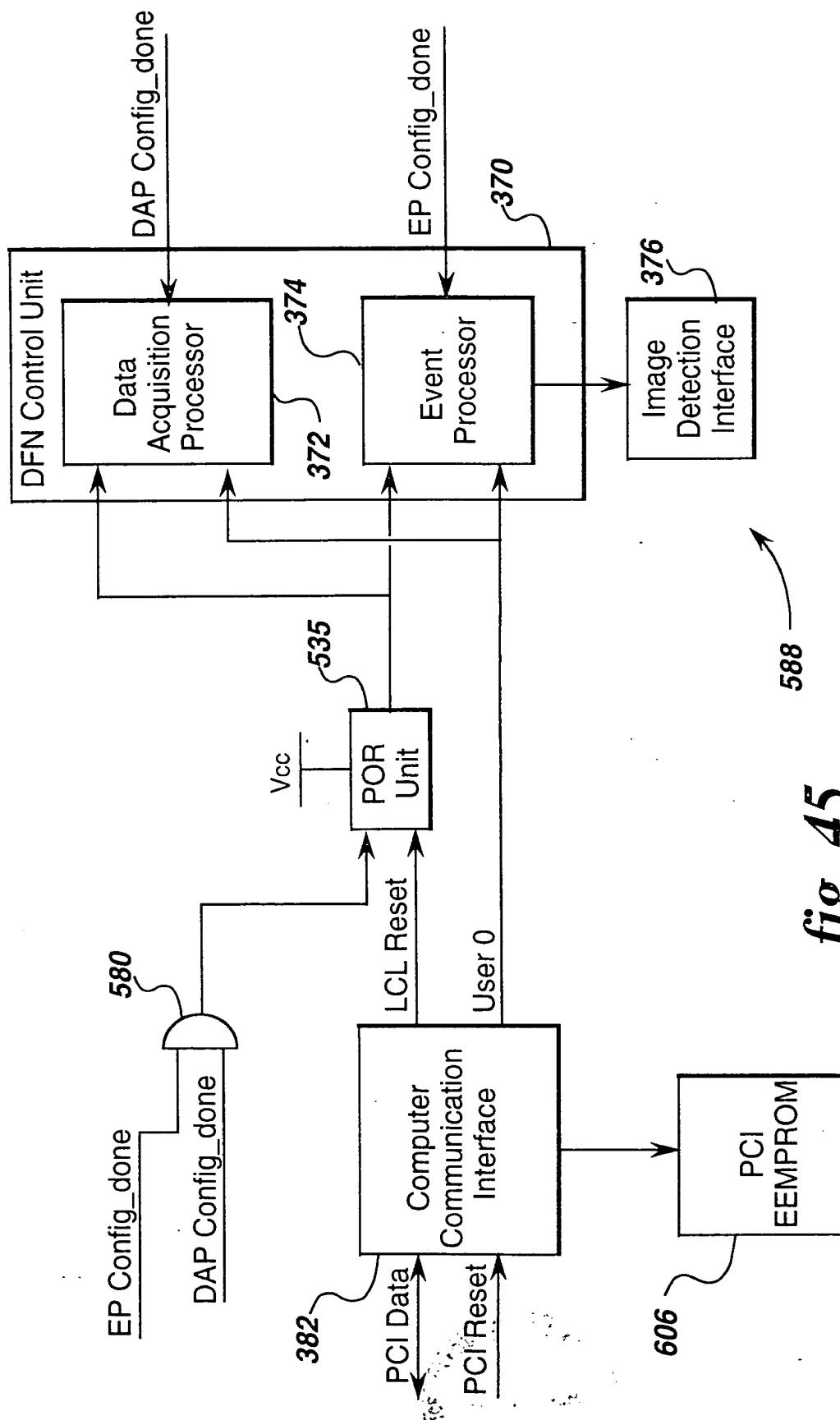


fig. 45

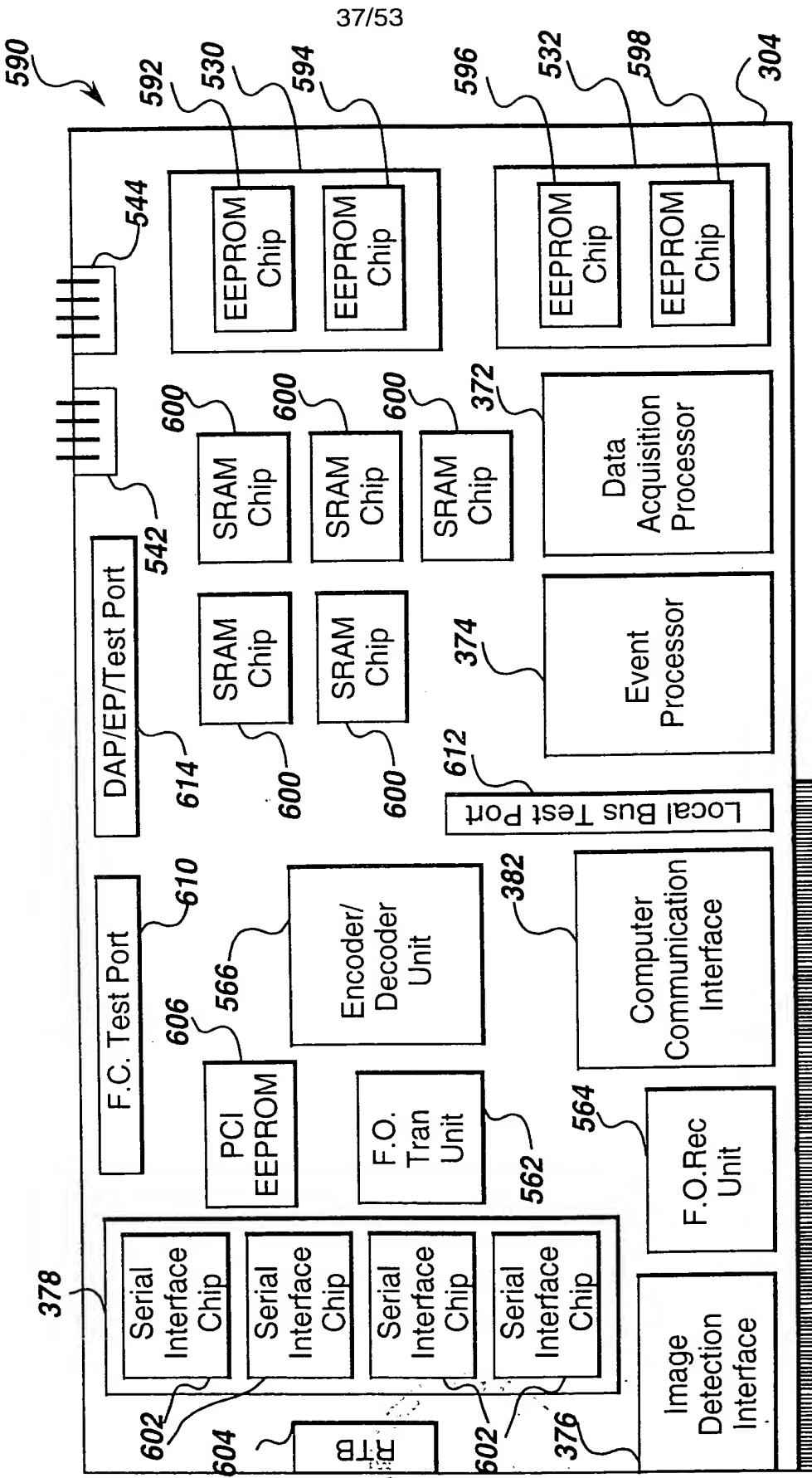
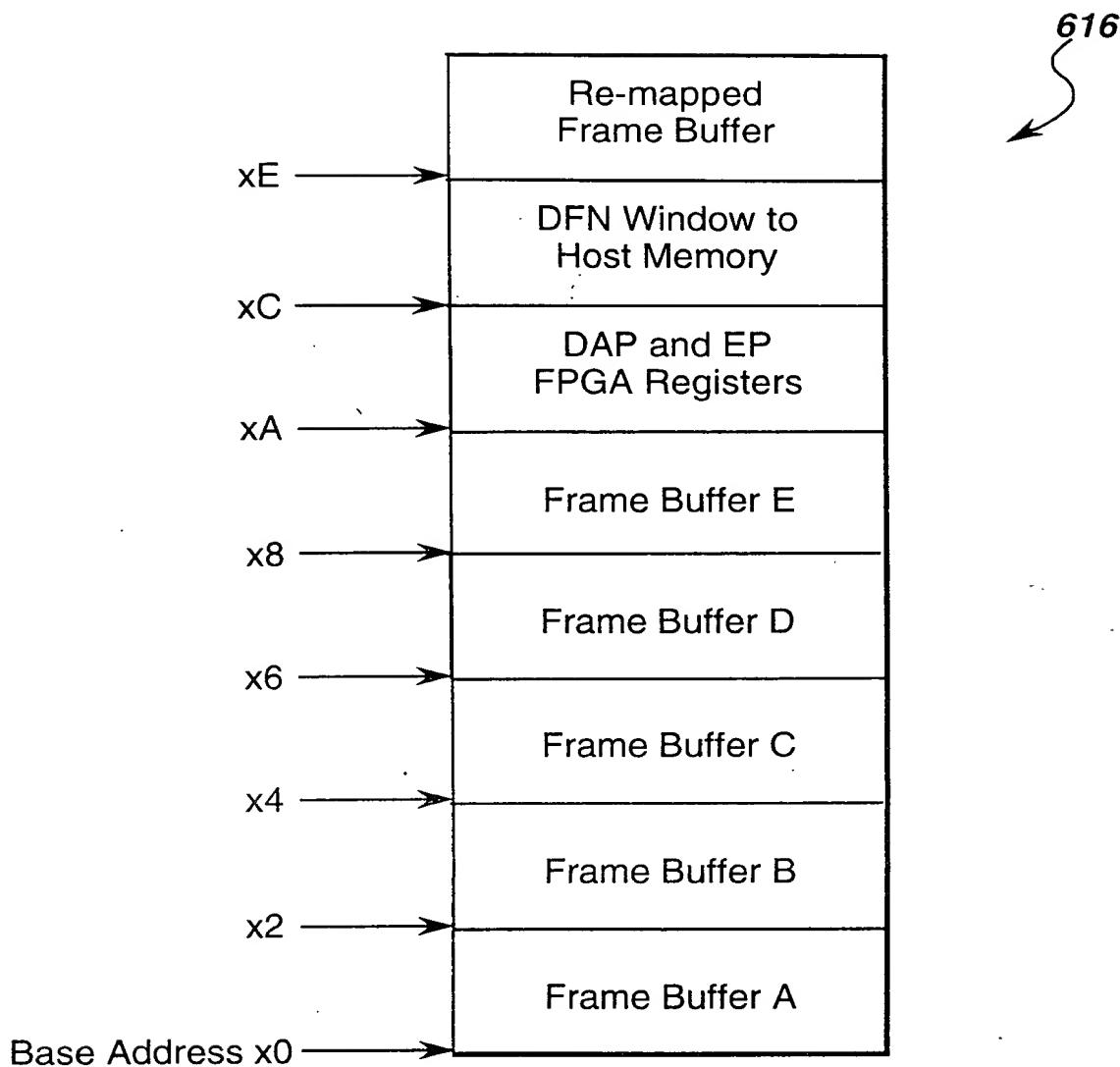


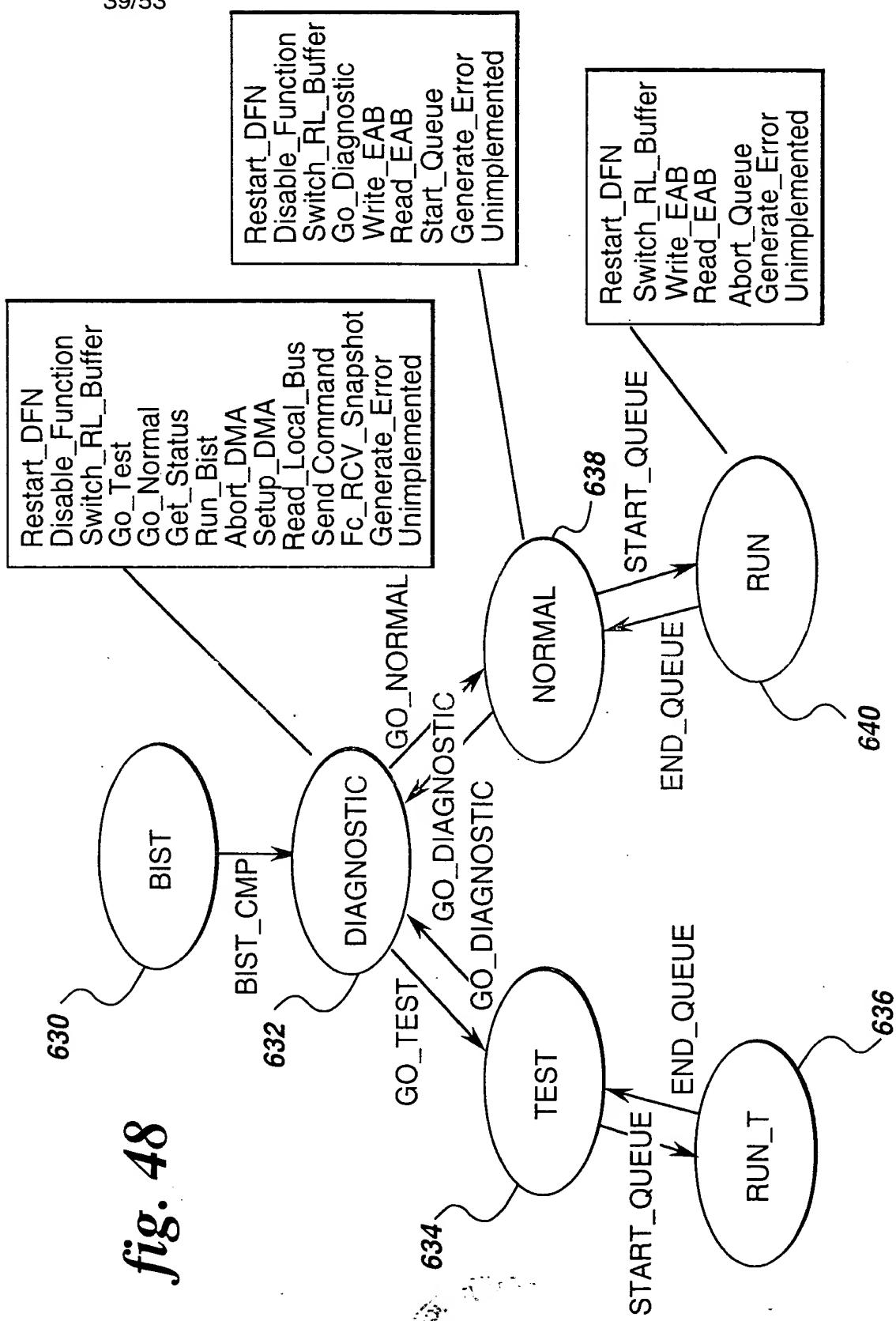
fig. 46



Mapping of 16 MByte PCI Address Space

fig. 47

fig. 48



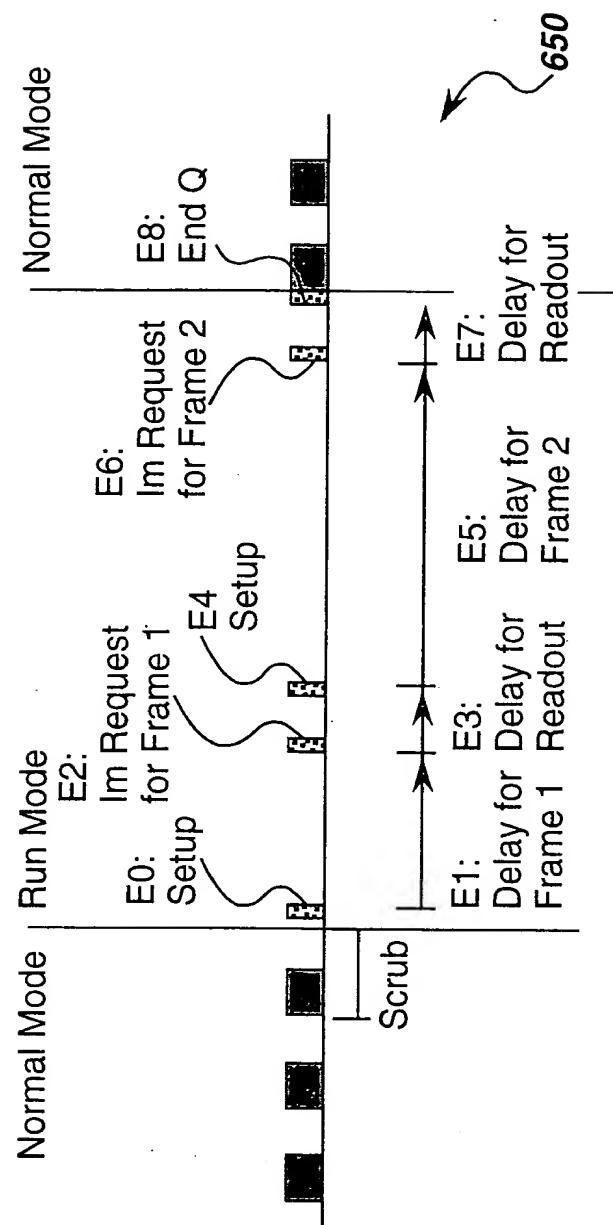


fig. 49

Event Mnemonic	Event (showing size of arguments)	Op Code (hex)	Data (bytes)	Total (bytes)
Endq	Endq	14	0	1
Delay (T)	Delay (0xffff ff ff)	10	4	5
Send (command, value)	Send (0xffff ff ff, 0xffff ff ff)	04	8	9
LoopKN (K, N)	LoopKN (0xffff, 0xff)	0C	3	4
LoopKF (K, F)	LoopKF (0xffff ff, 0xffff ff ff)	0D	5	6
Wait (F)	Wait (0xffff ff)	09	3	4
Flag (F)	Flag (0xffff ff, ff)	08	3	4

fig. 50

660



fig. 51

S1	S2								
SEND	<table border="1"> <tr> <td>HDR1(0)</td><td>HDR1(1)</td><td>HDR1(2)</td><td>HDR1(3)</td> </tr> <tr> <td>HDR2(0)</td><td>HDR2(1)</td><td>HDR2(2)</td><td>HDR2(3)</td> </tr> </table>	HDR1(0)	HDR1(1)	HDR1(2)	HDR1(3)	HDR2(0)	HDR2(1)	HDR2(2)	HDR2(3)
HDR1(0)	HDR1(1)	HDR1(2)	HDR1(3)						
HDR2(0)	HDR2(1)	HDR2(2)	HDR2(3)						

SEND

670

<i>Error Mnemonic</i>	<i>Description of Error</i>
FC_TIMEOUT	Timeout Expired With No ACK Detected
FC_BAD_ACK	ACK Did Not Match Transmitted Command
FC_EXTRA_ACK	Unexpected ACK Received
FC_EXTRA_CMD	New Send Event While Waiting for ACK From Previous Send
SIG_DETN	No Input Signal Power on Fibre Channel (Cable Disconnected?)
RXERROR	Fibre Channel Receiver Detected Bad Data (Defective Chipset?)
WRDSYNCN	Fibre Channel Data Link Unynchronized
CRXS(1)	Bad Received CRC Detected (Fiber-optic Cable Problem?)
CRXS(3) & CRXS(2)	Bad Order in Link State Machine (Defective Chipset?)

fig. 52

672

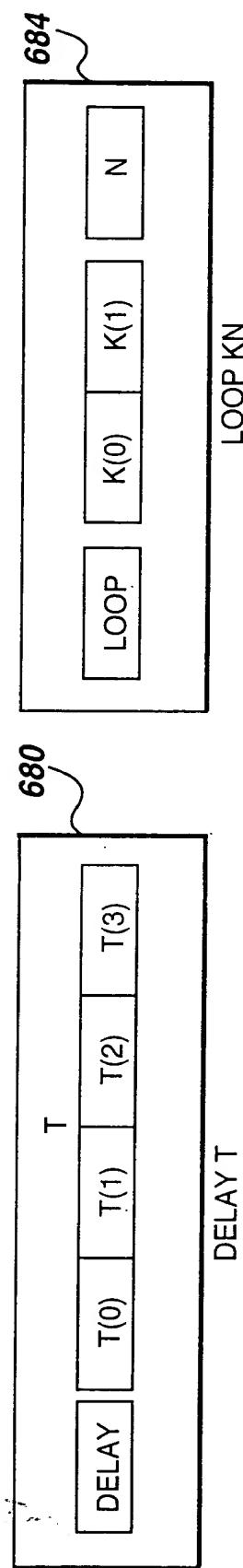


fig. 53

fig. 54



680

684

LOOP KN

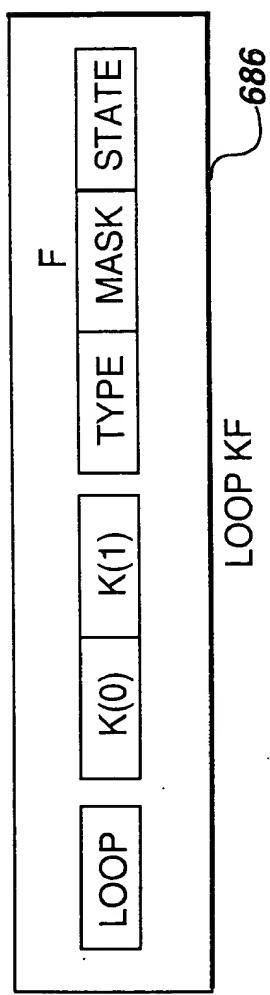


fig. 55

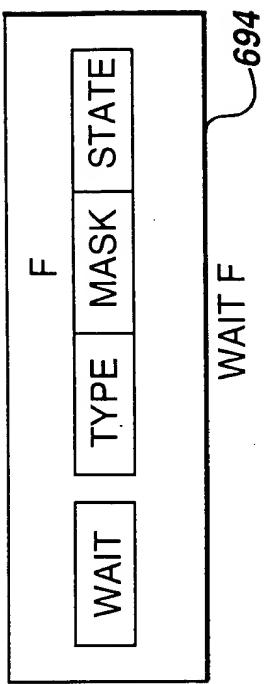


fig. 56

43/53

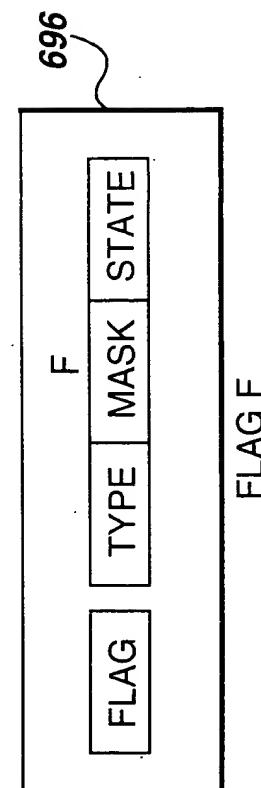


fig. 57

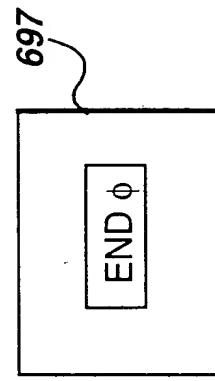


fig. 58

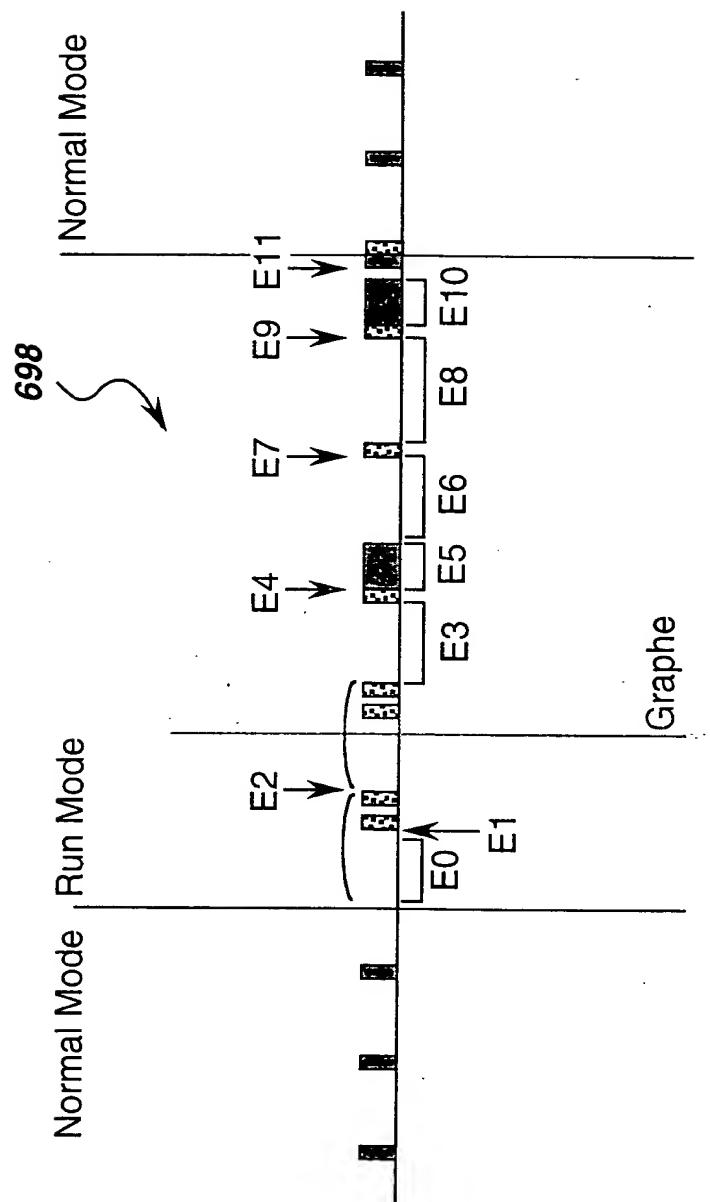


fig. 59

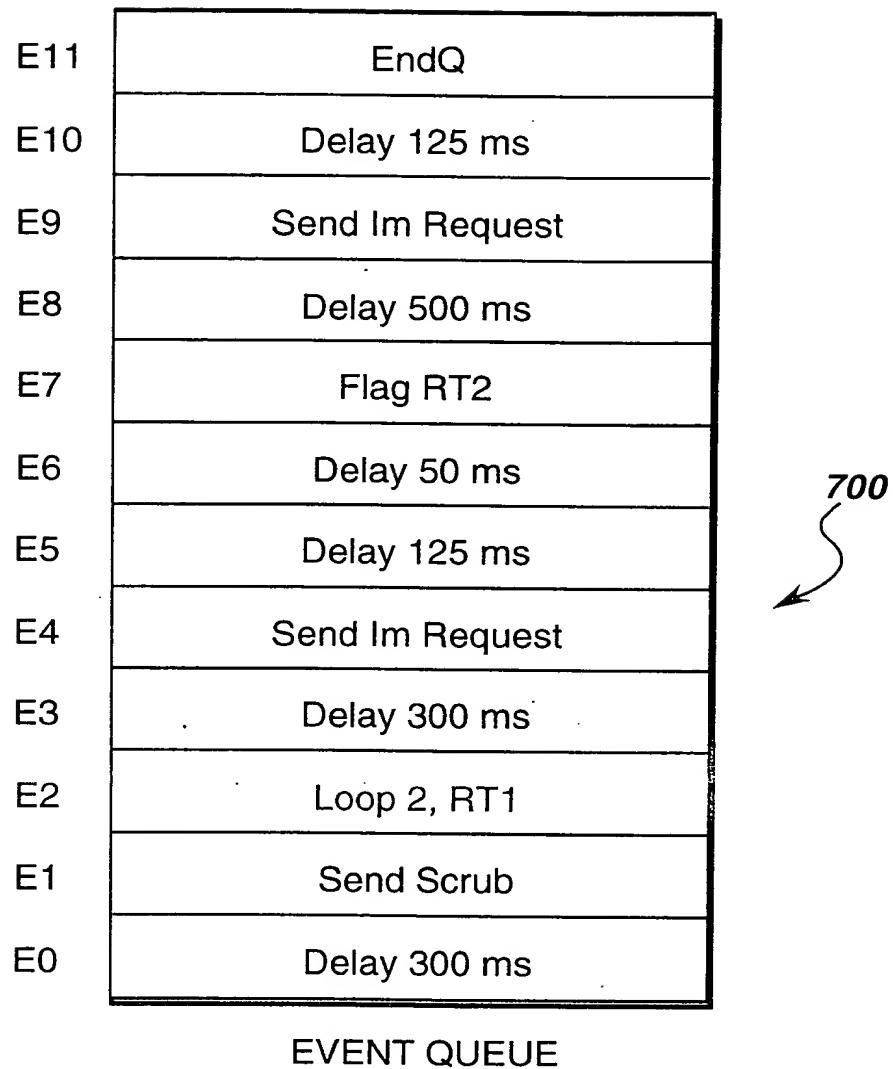


fig. 60

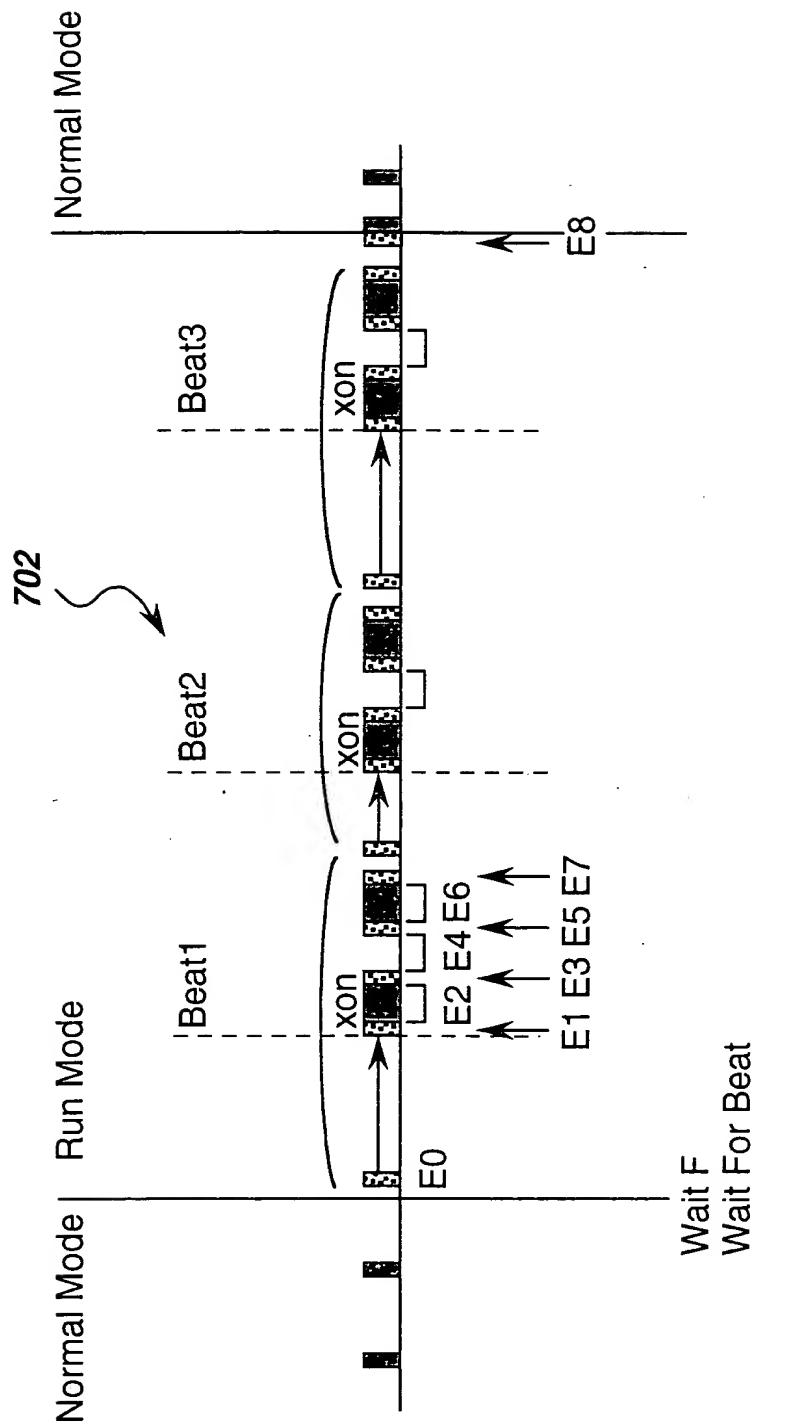


fig. 61

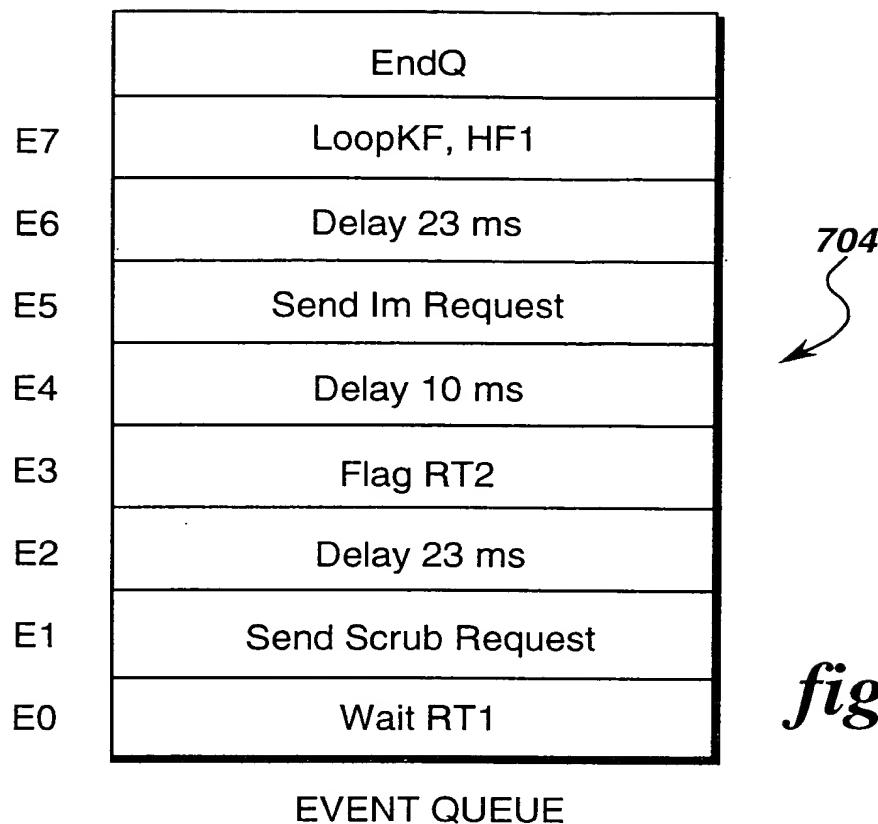


fig. 62

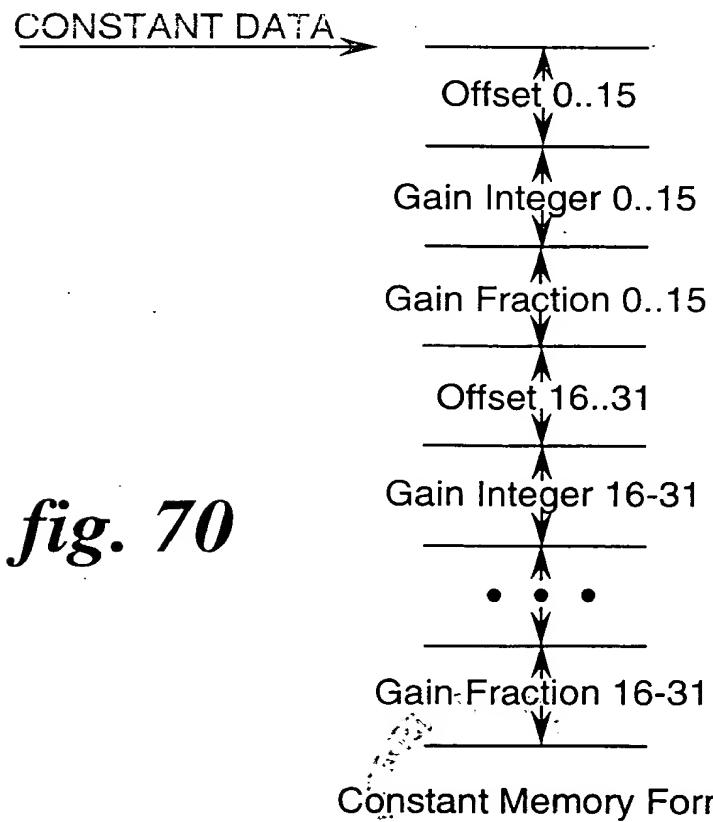


fig. 70

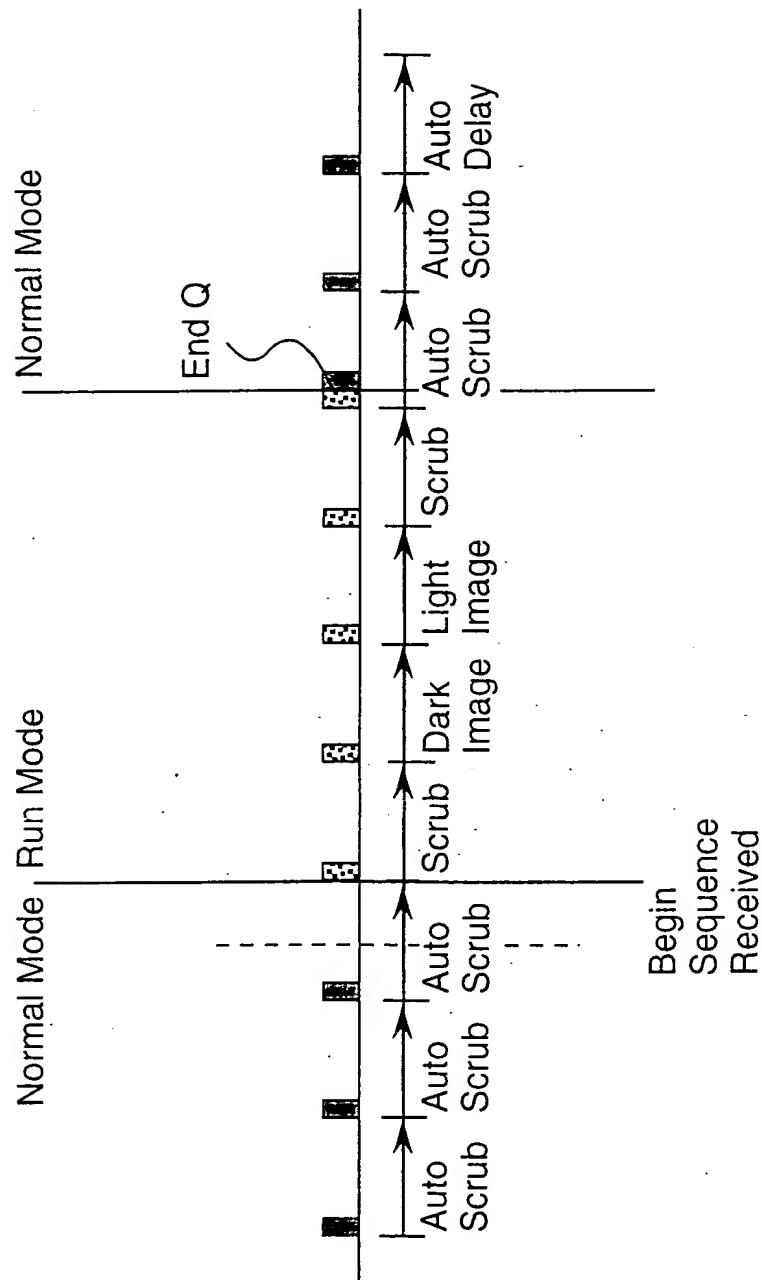


fig. 63

```
sequence_begin ();
# define qv defaults:
%qv1 =('delay_qv' => 5000),
# call frame with qv's
frame_type1(NULL, \%qv1, 1);
sequence_end ();
```

fig. 64

```
sub frame
{
$QVf = 'frame';
%qv = ('delay_qv' => [10000]);
%qp = ();
compile_init (@_, \%qp, \%qv, $QVf);
Delay('Delay_qv1');
compile_finit ();
}
```

fig. 65

49/53

```
pDFN->DFNChangeQueueVariable
(
(char *) SymName,           // variable name
(char *) sndBuf,            // new value
BufSize,                   // num bytes to write
(ULONG *) & Debug,          // developer info
);
```

fig. 66

```

// load and run the event sequence
pDFNBeginSequenceNoMappingNoLog
  (snum, "d:\HF.bin");
// assign data to be passed
sndBuf = 25000;
// change the queue variable
pDFN->DFNChange QueueVariable
  (
    (char *) SymName,           // variable name
    (char *) sndBuf,           // new value
    (ULONG ) sizeof sndBuf,    // num bytes to write
    (ULONG *) & debug,         // developer info
  );

```

```

sub frame_type1
{
  $HFfrm = 'frame_type1';
  %qv = ('delay_qv' => [20000]);
  %qp = ( );
  $image_cmd = [0x8000000,0x0];
  compile_init (@_, \%qp, \%qv, $HFfrm);
  Send ( $image_cmd);
  Delay('delay_qv');
  LoopKF(2, 0xAFFF01);
  compile_finit();
}

```

fig. 67

fig. 68

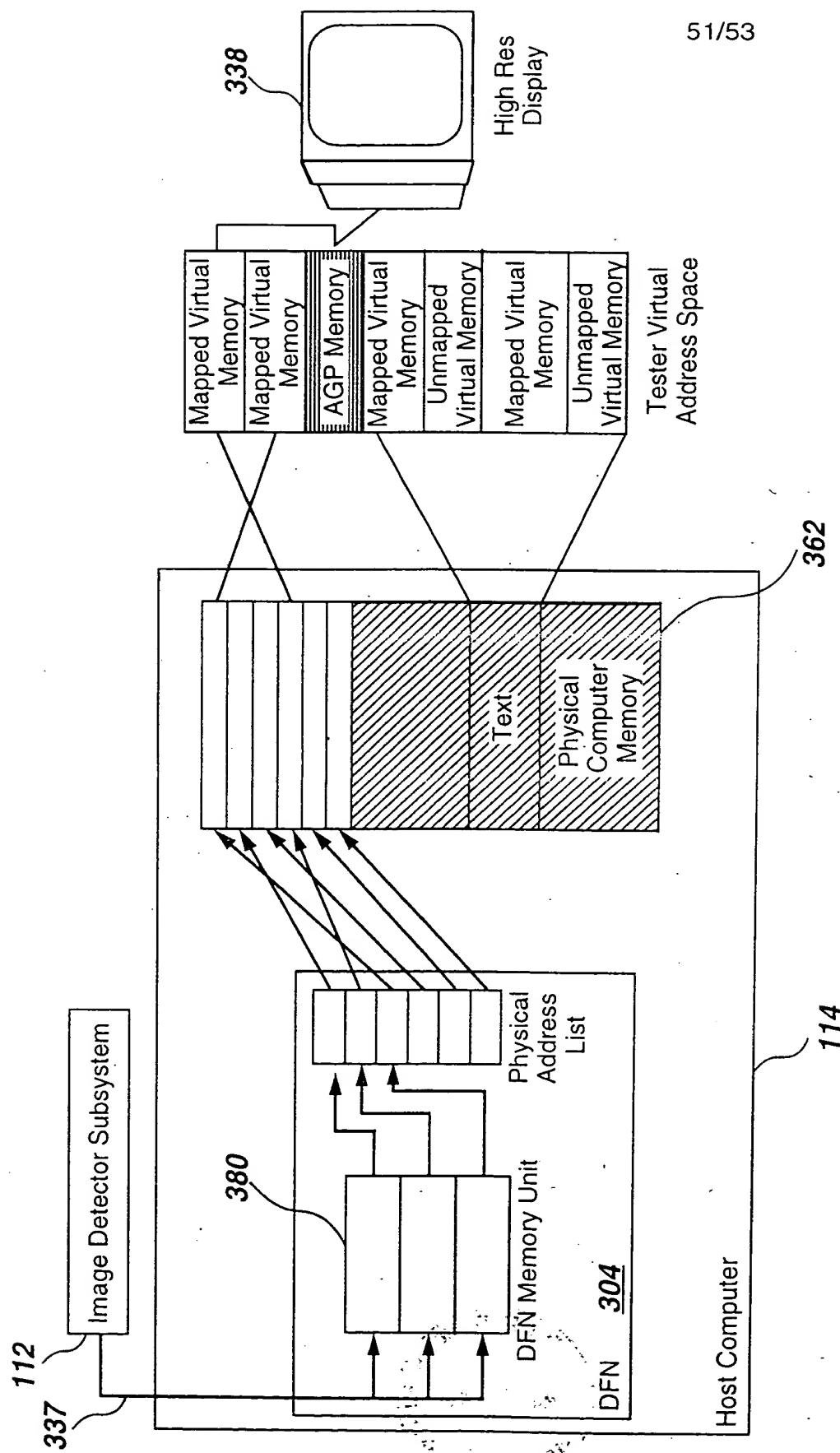


fig. 69

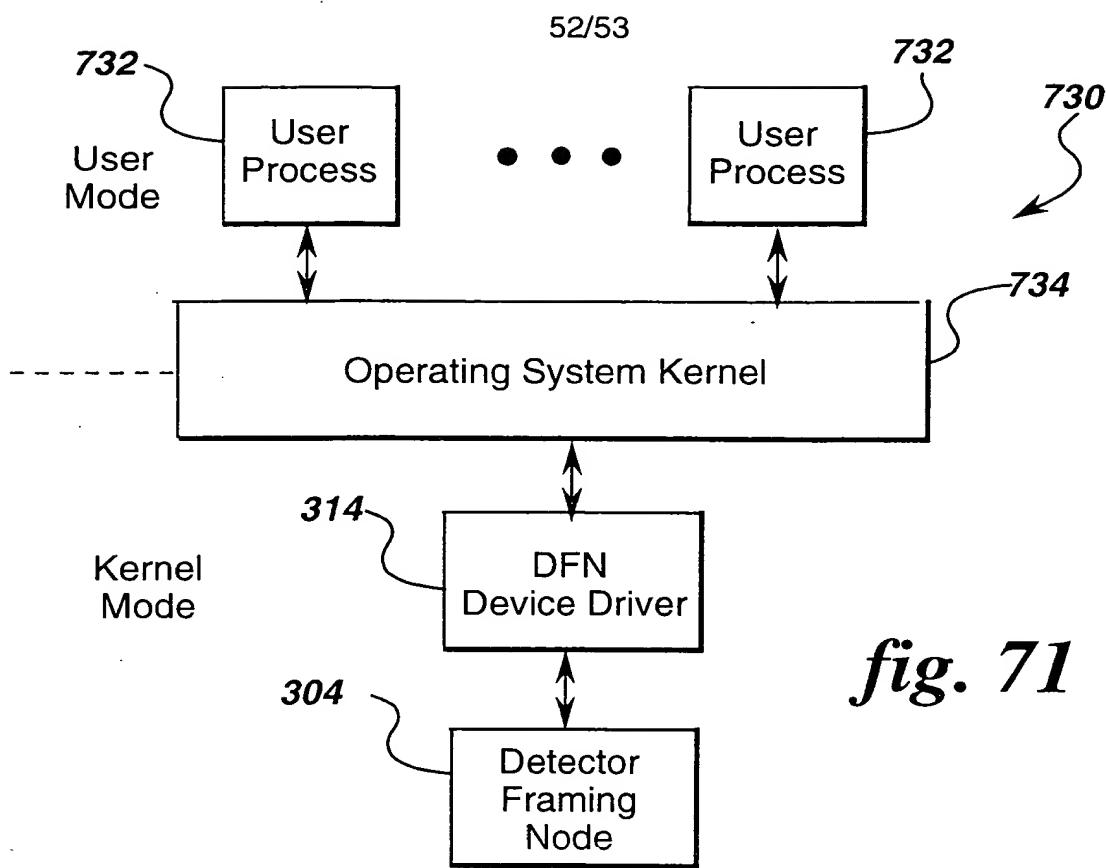


fig. 71

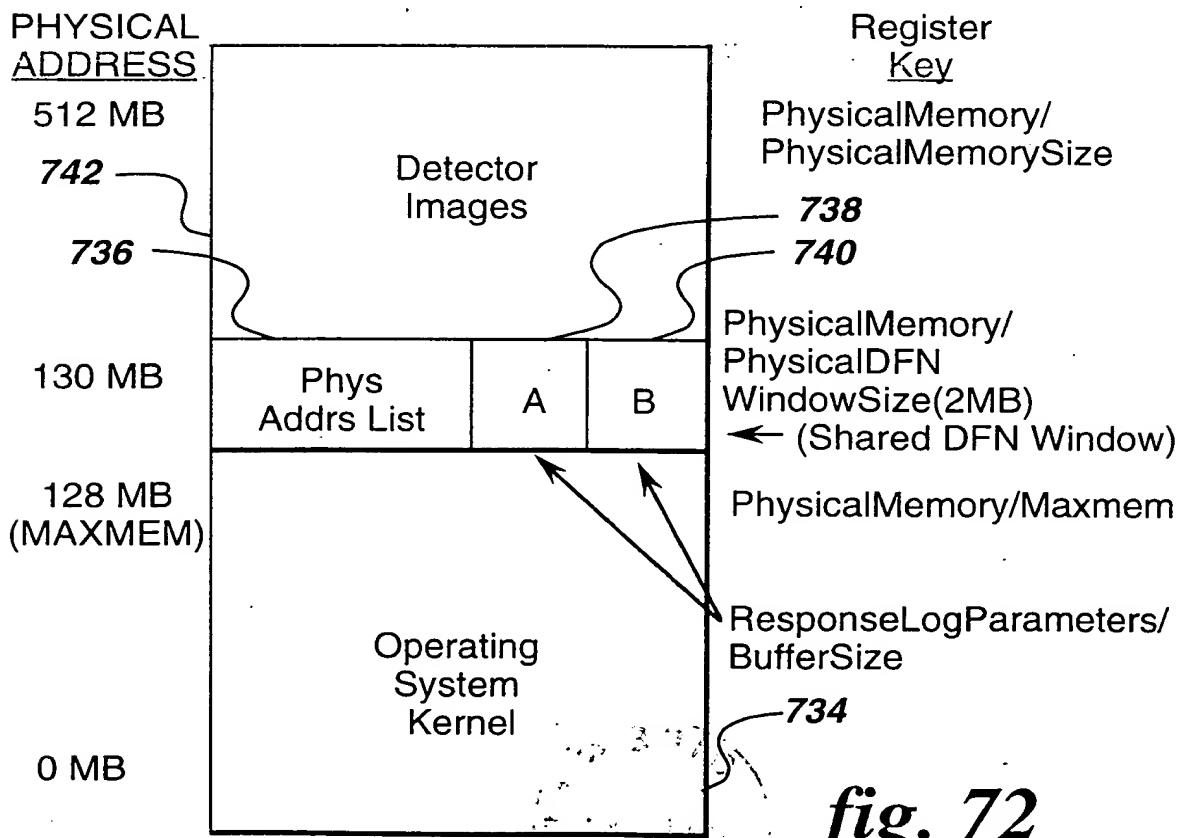


fig. 72

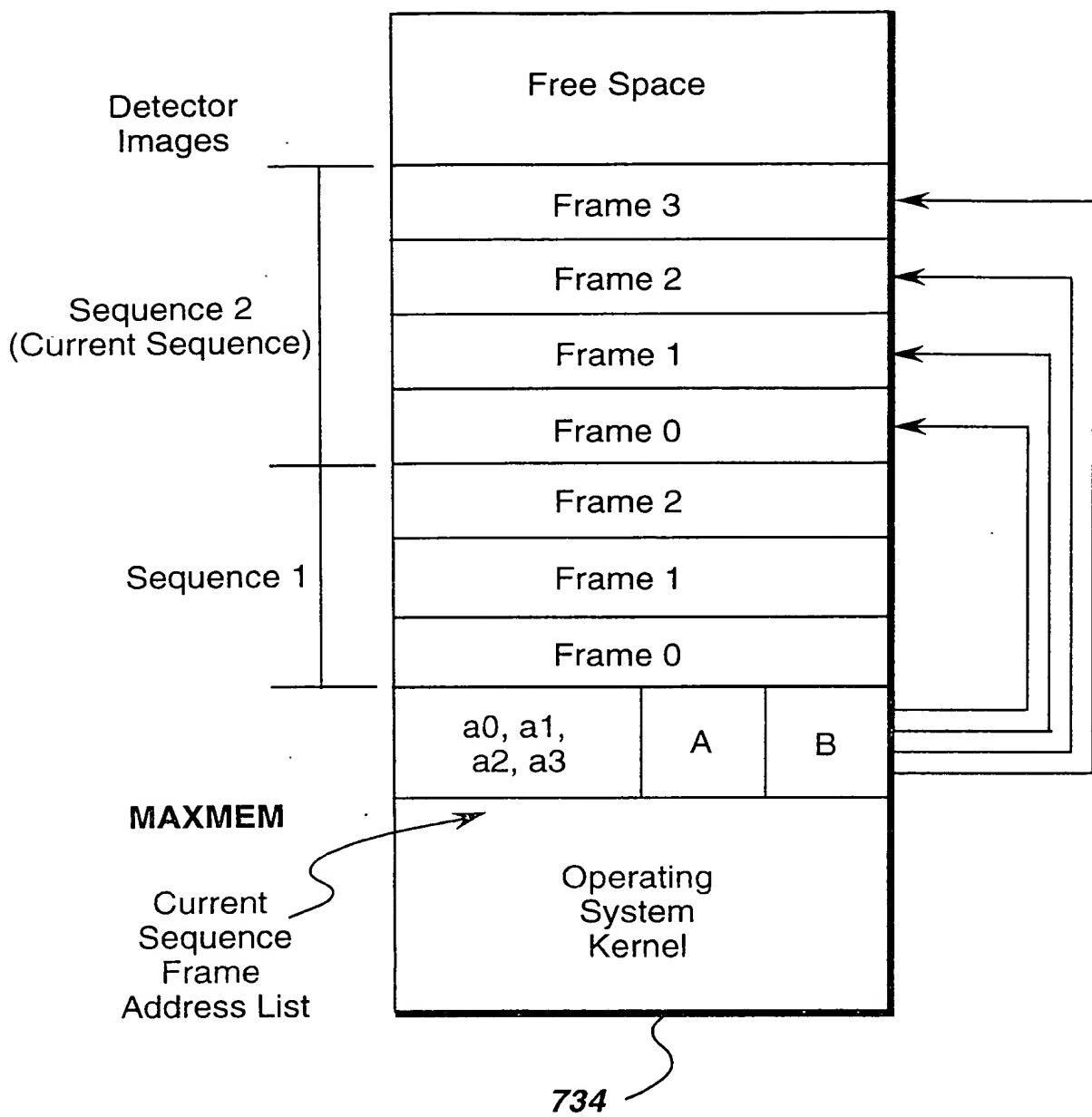


fig. 73